Ára: 30 Ft

μ mikro számítógép magazin

PROGRAM AIDS?

1988/8

ENTERPRISE Soft list '88.

1. (55) Magic Ball
2. (1) Sorcery
3. (49) Batman
4. (50) Airwolf
5. (2) Beach Head
6. (56) Enter Ball
7. (6) Star Strike 3 D
8. (4) Race Ace
9. (3) Wriggler
10. (7) Nodes of Yesod
11. (5) „Áttörés"
12. (8) Rablo Roulette
13. (51) Supr Pipeline II.
14. (9) Wizard's Lair
15. (11) Orient Express
16. (10) Chess (CYRUS) II.
17. (13) Centrum Ball
18. (14) Eggs of Death
19. (0) Core Wars
20. (12) The Abyss
21. (15) „Túrbó Rudi"
22. (16) Beatcha
23. (17) English is Fun I.—II.
24. (20) Fantasia Diamond
25. (23) Jack's House of Cards
26. (22) Steve Davis' Snooker
27. (18) Devil's Lair
28. (19) Mirror World
29. (21) Games Pack II.
30. (52) Fantomas
31. (53) Nautilus
32. (24) Chains
33. (26) Morden's Quest
34. (25) Land of Havoc
35. (30) King of the Castle
36. (29) Heathrow ATC
37. (28) Spanish Gold
38. (27) Playground Adventure
39. (54) Jammin
40. (31) Wordhang
41. (33) Happy Letters
42. (32) Happy Numbers
43. (34) Castle of Dreams
44. (35) Adventure Addie
45. (36) Market
46. (37) Newton almája
47. (38) Forma—1 autóverseny
48. (39) Digi-Spider
49. (40) Reversi-Dama — Awari
50. (41) Wind-Surfer
51. (42) Enter Stack
52. (43) Five in a Row
53. (44) Animal—Vegetable—Mineral
54. (45) Biorithm
55. (0) Passians—Black Jack
56. (0) Hungarian Demo I.—II.
57. (59) Grid Trouble
58. (64) Opening Strategies I.—II.
59. (60) Dot Breaker — Dot Collector
60. (61) Colossal Adventure
61. (62) Space Pirat
62. (58) Tombs of Doom
63. (0) „Körmöci" arany
64. (57) German is Fun
65. (63) Vokabel Trainer — Verben Trainer
66. (0) Babayaga
67. (0) Insent
68. (0) Games Pack III.
69. (0) Spion
70. (0) „soew" Voice Generátor
71. (0) Games Pack I.
72. (0) Spanish is Fun
73. (46) U-Boat (submarine simulator)
74. (47) Középfokú német nyelvvizsga I.—II.
75. (48) Középfokú angol nyelvvizsga I.—II.
76. (—) Juke Box
77. (0) War Ball
78. (—) Emerald Isle
79. (—) Dungeon Adventure
80. (—) Seventh Seal
81. (0) French is Fun
82. (—) Rocket Roger
83. (—) Dr. Stranloop
84. (—) Artist
85. (—) Lord of Time
86. (—) Snow Ball
87. (—) Return to Eden
88. (0) Russian is Fun
89. (0) Dictator
90. (—) CADCOM Warriors
91. (—) Falcon Patrol II.
92. (—) Frank Bruno's Boxing

6. ÉVFOLYAM
1988/8. SZÁM

mikro számítógép magazin

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG LAPJA

A szerkesztőbizottság vezetője:
Kovács Győző

A szerkesztőség munkatársai:
Babos János (tervezőszerkesztő)
Bakos Tamás (programozástechnika)
Broczkó Péter (hírek)
Kovács Győző (levelezés)
Petróczy Judit (könyvek)
Pinke György (NJSZT, alkalmazások)
Simonyi Endre
Szebenszki Sándor
Szulyovszky Csaba
Tamásné Lakó Erika
Terebessy Ákosné
Vizessy Mária

Felelős szerkesztő:
Könyves Tóth Pál

Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250

Levélcím:
1371 Budapest
Pf. 433.

Kiadja az Ifjúsági Lap- és Könyvkiadó Vállalat

Felelős kiadó:
dr. Király G. István
igazgató

Kiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660

Terjeszti a Magyar Posta
Előfizethető a hírlapkézbesítő hivataloknál
és a Posta Hírlap-előfizetési és Lapellátási Irodáján
(1900 Budapest XIII., Lehel u. 10/A)
vagy átutalással a 215—96 162 pénzforgalmi jelzőszámra.

Megjelenik havonta.
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti a Kultúra,
1389 Budapest, Pf. 149.
és a Magyar Média
1932 Budapest, Pf. 279.
86—0253

Szikra Lapnyomda
Budapest (88—0980)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

Címképünk:
Kiss Ilona munkája

Tartalom

Ismét: copyright 2
Feladatok — megoldások 7
Rendszerfejlesztési eszközök 21
Halálos vírusok? 23
Mit tud a QUALIGRAPH? 28
Játsszunk komolyan! 30
Egy sarokkal olcsóbb! 32
Közprogramok 33
Merre tart a világ? 34
µINFORM 39
Olvastunk . . . 40
Programtermék 45
Adok — veszek — cserélek 46

ISKOLA-SZÁMÍTÓGÉP

TechnoMIR 3
Magyarítsuk a BASIC-et! 4
3-MEG-1-gyel kényelmesebb 6

DIÁKROVAT

Pattogó C Plus/4-re 8
TOP-lista 8
Külső audio jel kezelése 9
Az Enterprise kazettás tárolása 9
Analóg óra Plus/4-en 10
Ha egy háziállattal több van 11

PROGRAMOZÁSTECHNIKA

BASIC és gépi kód 12
A Precomp Plus előfordító program III. 14

ENTERPRISE

Rajzolóprogram IS-BASIC-ben 24
Megkérdeztük az Enterprise-ról 25
Memória, grafika, billentyűzet 26
Mi a manó? 27

µPROGRAMOK

Spectrum—Loader 37

µKLUB

Ki ad magyarázatot? 38
Jelátvitel két PTA 4000 között 38

SAKK

A KGYK típusú gyalogvégjátékok 42

AZ OLVASÓ ÍRJA 43

KÖNYVEK 46

HÍREK, ÉRDEKESSÉGEK 47

PONTVADÁSZAT 48

Ismét: copyright

„Ne lopj!"
(Isten VII. parancsolata)

A másik rovatomhoz érkező olvasói levelek jelentékeny része egyre többet foglalkozik a szoftverlopással. A levélírók túlnyomó többsége általában elítéli az illegális másolást és a pénzért való terjesztést, de nincsenek kevesen azok sem, akik rendkívüli módon próbálják bizonygatni, hogy adott esetben (az ő esetükben) az, amit csinálnak, csak látszólag olyan, mintha megsértenék más szoftverfejlesztők szerzői jogait. Valójában ők nem követnek el semmiféle jogsértést, legfeljebb ügyesen kihasználják a törvény adta lehetőségeket.

Ezekből a történetekből, azt hiszem, legalább tíz adásnyi „Jogi esetek" összegyűlne. S talán még a legokosabb jogászokat is zavarba hozná egy-egy olyan csalafinta ügy, amit munkatársaim, de nagyon sokszor a µM olvasói is elmesélnek.

Nem is olyan régen egy kollégával utaztam együtt hazafelé, aki elmondta, hogy néhány kiváló szakemberrel együtt jól menő magánvállalkozásba kezdett. Most az az üzlet, hogy részben a hírhedt és eléggé el nem marasztalható huszonötezres törvényt próbálják megkerülni. Mégpedig úgy, hogy négy-öt darabban hozzák be a PC XT-ket és AT-ket, melyeket azután a kuncsaftnál raknak össze. Ily módon egy-egy számítógép alig kerül 20-25 százalékkal többe, olyan, mintha egyben hoznák be és adnák el valamelyik közintézménynek. Ráadásul még külön jövedelemhez is jutnak, amiért persze fájjon a törvényt hozó illetékes feje. Másrészt pedig szoftvert készítenek, elsősorban a már említett PC kompatibilis gépekhez.

Tovább beszélgetve kiderült, hogy van egy sikeres programterméküK, amit igencsak tetemes fejlesztőmunkával készítettek, és ami egy másik cég által, ki tudja milyen módon behozott és terjesztett (legálisan vagy illegálisan, ezt nem lehetett kideríteni) programcsomaghoz illeszkedik úgy, hogy annak szolgáltatásait bővíti, növelve egyben a hatékonyságát is.

Az is kiderült, hogy a fejlesztési ráfordítás nagy részére azért volt szükség, mert „fel kellett törni", azaz meg kellett fejteni az eredeti szoftvert, hiszen addig nem tudtak hozzálátni a kiegészítő program megírásához, amíg a „felspécizendő" programot forrásmélységig nem ismerték.

Miután az eredeti programot az a bizonyos másik cég már igen sok helyre eladta, így most a beszélgetőtársam új cége könnyű helyzetben van. Megkeresi ugyanis mindazokat a felhasználókat, akik az eredeti programot megvették, és nem túl drágán felajánlja nekik a teljesítmény- és szolgáltatásnövelő kiegészítést, ami – teljesen nyilvánvaló módon – az ő cégének a szellemi terméke. Még azt is megtudtam, hogy mielőtt ezt a programterméket forgalmazták volna, megkérdeztek egy jogászt is, nem törvénybe ütköző-e, amit csinálnak. A válasz az volt, hogy nem.

Elgondolkoztam a dolgon, merthogy a természetes jogérzékem már a történet elmesélése alatt is berzenkedett az ilyen programfejlesztés és nemkülönben a forgalmazás ellen. Véleményem szerint ugyanis útitársam cége már akkor elkövette az első törvénysértést, amikor az eredeti szoftvert engedély nélkül megfejtette. Tudniillik csak a program ismeretének birtokában volt képes a fejlesztői csoport a kiegészítő programot elkészíteni, és így az eredeti program – ami másnak a szellemi tulajdona – szolgáltatásait és hatékonyságát megnövelni. Máskülönben ez az új termék önállóan, tehát az eredeti program nélkül semmire sem használható, csak azoknak jelent értéket, akik az eredeti programnak is a tulajdonosai.

Útitársam cége, mint mondta, arra is gondolt, hogy megpróbálják a terméküket külföldön, tőkés piacon is értékesíteni. Én erről a szándékról élénken igyekeztem lebeszélni őt, mondván, hogy ott az eredeti program tulajdonosa egy pillanat alatt észreveszi szellemi tulajdonának megrövidítését, és akkor semmi sem menti meg őket egy pertől, amit – szerintem – egész biztosan elveszítenek.

Vitánk döntetlenül végződött. Érvelésemmel nem tudtam meggyőzni útitársamat, mert úgy érezte, hogy itt semmiféle jogsértés nem történt. Azt a javaslatomat pedig, hogy keressék meg az eredeti program külföldi tulajdonosát és ajánljanak fel neki egyezséget, fizessenek például royalty díjat minden eladott programjuk után, a partner programjából felhasznált szellemi érték mértékében – nevetségesnek tartotta.

A beszélgetés után elgondolkoztam azon, hogy miért vagyunk mi ilyen – mondjuk ki – lopósak. Miért ítéljük el azokat, akik ellopnak egy ezüstkanalat, de szemrebbenés-nélkül tudomásul vesszük, sőt néha még dicsőségnek is tartjuk, ha valaki egy sokévi fejlesztési munka eredményét tulajdonítja el. Sőt még áruba is bocsátja, ráadásul nemcsak az elmondotthoz hasonló, kissé bonyolult és becsapós esetben.

Az elmúlt hónapokban sok olyan megbeszélésen vettem részt, ahol külföldi szoftver magyarországi forgalmazásáról volt szó. Minden alkalommal megkérdezték a külföldi partnerek, hogy milyen másolás elleni védelmet tudunk ajánlani, mert hallották, hogy nálunk egyetlen eladott szoftver sincs „biztonságban": azt nemcsak az amatőrök, de még a professzionális szervezetek is másolják és továbbadják.

Azt hiszem, szükségtelen ecsetelnem, hogy a hazai szoftver exportjának erősödésében mennyire nem mindegy, milyen a „hírünk a világban". Hiszen ahhoz, hogy exportálhassunk, importálnunk is kell, mégpedig elsősorban olyan technológiai termékeket, amelyekből a hazai szoftverfejlesztés teljesítménye fokozható és egyben a piacképessége is növelhető.

Azt hiszem, valamit tennünk kell. Tovább már nem halasztható annak az ellenőrzése, hogy a hazai piacon mind a magán, mind a közületi szférában csak olyan termékeket forgalmazzanak, amelyeknek copyrightja egyértelműen tisztázott. Egy ilyen „tiszta" piac nemcsak a külföldi megítélésünk szempontjából lényeges, hanem igazságosabbá teszi a jövedelmelosztást is. Nem nehéz belátni, hogy egy lopott szoftvertermékeknek az „adaptálása" – az esetek nagyobbik részében – sokkal kevesebb munkával jár, mint egy termék teljes kifejlesztése, ami azután a tol . . . – pardon – az adaptálónak csinos és persze meg nem szolgált jövedelmet hoz. Ma a kormányzat nagyon odafigyel arra, hogy megadóztassa a láthatatlan jövedelmeket, így tehát elvileg a mások szellemi termékeinek eltulajdonításából származó jövedelem is adózik – ábban is biztos vagyok, hogy nem is keveset –, de ezzel a problémát még nem oldottuk meg. Sőt bizonyos mértékben legalizáltuk a helyzetet, pontosan úgy, mint az orvosoknál a hálapénzt, hiszen mindenki tudja, hogy az adóztatástól a beteg nem gyógyul meg gyorsabban, és az orvosoknak sem lesz annyi jövedelmük, amennyit a munkájuk után megérdemelnének.

Pedig itt is és ott is ez volna a cél!

KOVÁCS GYŐZŐ

TechnoMIR

Alkalmazási példák I.

Sorozatunk befejezéseként két részben a rendszer alkalmazására mutatunk be példákat. Ha ezen gondolkodunk, hagyatkozhatunk a fantáziánkra, hiszen alig akad mindennapjaink gyakorlatában olyan terület, ahol ne hasznosíthatnánk személyi számítógépünket és az ezt kiegészítő TechnoMIR interfészrendszert. Tanáremberek lévén, elsősorban az oktatásból vesszük példáinkat. Mégpedig olyanokat, amelyek továbbgondolkodásra késztetnek, esetleg adaptálásra is alkalmasak.

Programozható áramkörök

Az elektronikus áramkörök oktatásában kiválóan alkalmazható a DOUT modul a különböző programozható áramkörök működésének demonstrálására. Az *1. ábra* egy ilyen áramkört mutat be. Ez a műveleti erősítős alapkapcsolás a relék állapotától függően 10:1, 1:1, 1:10 arányú feszültségerősítést, illetve osztást tesz lehetővé. A Bolyai János Híradástechnikai Szakközépiskolában ezt úgy fejlesztették tovább, hogy a programozható áramkört is egy komplex rendszerből valósítják meg. Így kapják a KVM-02/Elastique komplex villamos mérőrendszert, amely a tanulók villamos és villamos jellegű méréseinek univerzális gyakorló eszköze. E két rendszer közös alkalmazását mutatja be a *3. ábra*.

Programozható műszerek

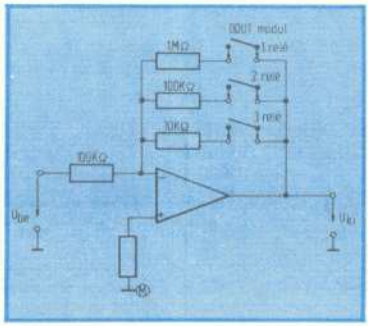
Az ilyen jellegű alkalmazás mintegy folytatása is egy kicsit az előzőekének. Itt a legjelentősebb elem az A/D modul lesz, hiszen szinte valamennyi digitális műszer alapáramköre az analóg-digitális átalakító. Érdemes emlékeztetőül visszalapozni a Magazin 1988/6. számához, ahol sorozatunkban ezt a modult mutattuk be.

A *2. ábrán* látható a TechnoMIR modulokból megvalósítható digitális multiméter (több elektromos mennyiség mérésére alkalmas műszer) elvi blokkvázlata. Az ábrán látható az A/D és az AMUX modulokon kívül egy olyan modul is, amelyet sorozatunk keretében kisebb jelentősége miatt részletesen nem mutatunk be. Ez az UNI modul. Az univerzális építő modullal olyan feladatokat is elvégeztetünk a számítógéppel, amelyek megoldására sorozatgyártásban nem készült modul. Egyaránt alkalmas egyéni elképzelések megvalósítására és speciális modulok kivitelezésére. A modul tartalmaz címdekódert is, de maga a modulcím nincs előre meghatározva. Ezt a tizenhat meglévő címből a felhasználó választhatja ki.

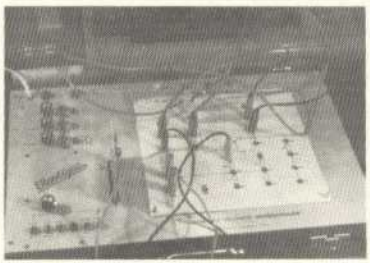
Ebben az UNI modulban benne van mindaz az átalakító típus, amely különböző villamos mennyiségek méréséhez szükséges. Ezeknek az átalakítóknak közös tulajdonsága, hogy a mérendő villamos mennyiséget arányos egyenfeszültséggé alakítják át.

Az AMUX modulnak az a feladata, hogy kiválassza és az A/D modulhoz eljuttassa a mérendő jelet. Az A/D modulban pedig létrejön az átalakítás, a konverzió az analóg mennyiség digitális megfelelőjére.

Ezt az alkalmazási példát továbbgondolva juthatunk el a következő területre.



1. ábra



3. ábra

2. ábra *4. ábra*

Automatikus mérőrendszerek

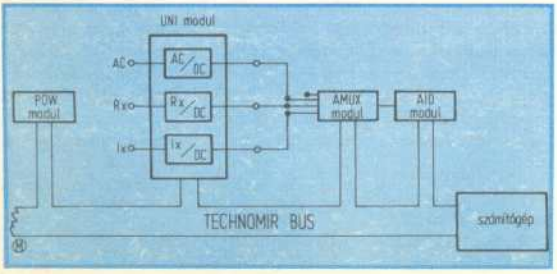
Az automatikus mérőrendszerek lényege, hogy a mérő személy szerepét a mérési folyamatban egy vezérlőegység – ebben az esetben személyi számítógépünk – veszi át. Ennek megfelelően az általunk megírt program szerint állíthatók be a műszerek, választhatók ki a mérési pontok, értékelhetők a mérési eredmények, mint ahogy eszerint működik egy adott algoritmus is.

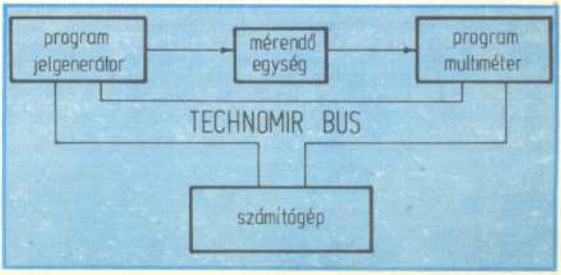
Az alkalmazási példa elvi blokkvázlata a *4. ábrán* látható. A programozható jelgenerátor egység talán nem szorul bővebb magyarázatra, hiszen az előzőekhez képest ennél csak annyi az eltérés, hogy D/A modult használunk. Ahhoz viszont felkészült programozónak kell lennünk, hogy a D/A modul kimeneti feszültsége valamely hullámformának megfelelően alakuljon az idő függvényében.

Természetesen itt mindenképpen igaz az a megállapítás, hogy a rendszer nem alkalmas preciz laboratóriumi mérésekre, de demonstrációs célokra annál inkább. Jól felhasználható tehát arra, hogy megfelelő program segítségével bemutassuk, miként lehet intelligens mérőrendszert létrehozni a kevésbé pontos hardvereszközökből.

Mihez kezdhetünk a gépben lévő mérési adatokkal? A mérési adatok feldolgozásától, ezek tárolásáig, dokumentálásáig széles annak a skálája, amit a TechnoMIR-t alkalmazva a ma működő mérőrendszerekről bemutathatunk. Nem is beszélve arról, hogy a demonstráción kívül a konkrét mérési eredményeket is felhasználhatjuk.

KIRÁLY LÁSZLÓ – ALBU LÁSZLÓ





Magyarítsuk a BASIC-et!

A számítógéppel ismerkedők első lépésként alapfokú tanfolyamokon többnyire a BASIC nyelvet és annak segítségével a programozás elemeit sajátítják el. Vannak — és általában nem a legfiatalabb korosztálybeliek —, akik nehezen veszik ezt az első akadályt (is), mert úgymond sem angolul nem tudnak, sem a matematika nem volt erősségük. Az ilyen alapvető félreértésből fakadó előítéletnek úgy is elejét vehetjük, ha az (angol) BASIC kulcsszó helyett magyar kulcsszavakkal programoztatunk. Eredendően magyar kulcsszavakat azonban csak a Homelab—3 gép értelmez.

A magyar kulcsszavak megfogalmazása amúgy is zsákutcába vezethet, mert vagy nagyon hosszú körülírást kell alkalmazni, ami — mivel a gépírás sem erőssége a kezdőknek — igen hosszadalmassá teszi a program írását, vagy rövidítéseket írunk be. Ha viszont már a szimbolikus nyelvet használjuk, ilyen jeleket tanulunk-tanítottunk meg, elengedhetetlen, hogy az legalább az illető gép saját BASIC szókészlete legyen.

Ennek ellenére a kezdeti idegenkedést feloldhatja a magyar kulcsszókészlet.

```
10        VEGYES BASIC EREDETI & !MAGYAR
12 POKE16561,0:POKE16562,125:CLEAR50
14 FORL=32001TO32072:READJ:POKEL,J:NEXT
16 FORL=25600TO25606:READJ:POKEL,J:NEXT
18 POKE16526,0:POKE16527,100:A=USR(0)
20 L=32080:    FORI=1TO60
22 READ A$:J=LEN(A$)+1:POKEL,J:L=L+1
24 FORZ=1TOLEN(A$):J=ASC(MID$(A$,Z,1))
26 POKE L,J:L=L+1 :NEXT:PRINT"*": NEXT
28 !FELEJTS
30 DATA 227, 62, 29,188, 32,  3, 62, 91
32 DATA 189,227,194,120, 29,205,120, 29
34 DATA 245,254, 33, 40,  2,241,201
36 DATA  17, 80,125, 14,  0,229,213, 26
38 DATA  71,197,  5, 19, 35, 26,190, 32
40 DATA  11,  5, 32,247,193,209,209,209
42 DATA  62,128,129,201
44 DATA 193,209,197, 72,  6,  0,235,  9
46 DATA 235,193, 12, 62, 60,145,225, 32
48 DATA 216,241,195,151, 25
50 DATA  33,  1,125, 34,  4, 64,201
52 DATA VEGE,CIKLUS,RAJZKI,RAJZBE,RADIR
54 DATA CMD,VELETLEN,KOVETKEZO,ADAT
56 DATA BEVITEL,HELY,OLVASS,LEGYEN,MENJ
58 DATA FUSS,HA,ELOLROL,ALPROGRAM
60 DATA VISSZA,MEGJEGYZES,ALLJ,KULONBEN
62 DATA KOVESS,NYOMKI,SZOVEG,EGESZ
64 DATA EGYSZERES,KETSZERES,LINE,JAVIT
66 DATA HIBA ESETEN,FELOLDAS,KIAD
68 DATA ELAGAZAS,NYIT,FI,GET,PUT,ZAR,LO
70 DATA MER,NEV,KILL,LS,RS,SAVE,BELEP
72 DATA NYOMTASS,DEF,DIREKT,IRJ,FOLYTAT
74 DATA TAR TARTALOM,PAPIRRA,ELHAGYNI
76 DATA SZAMOZZ,NULLAZZ,SZALAGROL
78 DATA SZALAGRA,FELEJTS
```

1. lista

Az itt közölt programokban csak annyi a megkötés, hogy a DATA sorokba ugyanolyan sorrendben kell beírni a magyar kulcsszavakat, mint ahogy azok a gép eredeti kulcsszótáblájában megtalálhatók. Hogy ezek a kulcsszavak milyen hosszúak, vagy hogy mennyire találó a megfogalmazásuk, az a programozó kénye-kedve szerint alakulhat. Így például a FOR helyett CIKLUS, CIKLUS INDUL, EGYMÁS UTÁN, SORBA, VÉGYE FEL . . . valamelyike írható a magyarított BASIC kulcsszótáblába.

HT—1080Z

Az *1. lista* a 80H—BBH tokenekkel helyettesített kulcsszavak magyarítását teszi lehetővé. Mód van a (D7H—FAH) függvények átírására is. Az olyan BCH—D6H tokenű kulcsszavak magyarítását, amelyeknek nincs önálló végrehajtó rutinjuk, a program nem tudja értelmezni, így nem is vizsgálja azokat.

A magyarított HT kulcsszavakat „!" jellel kell kezdeni. A végrehajtás során az utasításokat az RST 10 rutin felhasználásával az interpreter, a 32000—32072 dec címekre beírt rutin pedig a saját kulcsszótáblázatot vizsgálja.

A DATA sorokba tetszőleges, de eredeti BASIC kulcsszavakat nem tartalmazó meghatározásokat írhatunk magyar kulcsszóként.

Az eredeti BASIC kulcsszavak keverten is használhatók a mi új meghatározásainkkal, azzal a szintaktikai megkötéssel, hogy a magyar kulcsszavakat „!" jellel kell kezdeni. A másik, szintén HT—1080Z számítógépen alkalmazható program szintén az 10 rutin manipulálása után lehetővé teszi magyar kulcsszavak alkalmazását.

A program segítségével a HT 80H—D6H tokenekkel helyettesí-

2. lista

```
10  ***** MAGYARITOTT HT BASIC *****
11 POKE16561,235:POKE16562,116:CLEAR50
12 CLS:PRINT"TURELEM BETOLTES"
13 FORL=25600TO25606:READJ:POKEL,J:NEXT:L=29931
14 DATA33,236,116,34,4,64,201,227,62,26,188
15 FORI=0TO14:READJ:L=L+1:POKEL,J:NEXT
16 DATA 32,3,62,130,189,227,32,3,195,211,117
17 FORI=0TO15:L=L+1:POKEL,PEEK(13268+I):NEXT
18 FORI=0TO42:READJ:L=L+1:POKEL,J:NEXT
19 DATA0,0,0,0,0,0,0,0,0,0:REM EDIT!!!
20 DATA229,245,254,181,40,9,241,245,254,180
21 DATA 40,8,241,225,201,1,53,117,24,3,1,58
22 DATA 117,241,225,17,30,29,213,197,195,120,29
23 FORI=0TO156:L=L+1:POKEL,PEEK(11049+I):NEXT
24 POKE30069,139:POKE30070,117:POKE30111,150
25 POKE30112,117:POKE30141,185:POKE30142,117
26 POKE30158,196:POKE30159,117
27 POKE30135,0:POKE30136,122
28 L=L+1:POKEL,241:L=L+1:POKEL,205
29 L=L+1:POKEL,120:L=L+1:POKEL,29
30 FORI=0TO22:L=L+1:POKEL,PEEK(6786+I):NEXT
31 FORI=0TO206:L=L+1:POKEL,PEEK(7104+I):NEXT
32 POKE30245,255:POKE30246,121
33 POKE30206,137:POKE30207,118:POKE30212,165
34 POKE30213,118:POKE30216,171:POKE30217,118
35 POKE30224,137:POKE30225,118:POKE30231,137
36 POKE30232,118:POKE30241,137:POKE30242,118
37 POKE30249,107:POKE30250,118:POKE30271,60
38 POKE30272,118:POKE30287,103:POKE30288,118
39 POKE30363,250:POKE30364,117
40 FORI=0TO10:L=L+1:POKEL,PEEK(6812+I):NEXT
41 L=L+1:POKEL,195:L=L+1:POKEL,167:L=L+1
42 POKEL,26:L=31231:FORI=128TO204:READA$
43 L=L+1:POKEL,ASC( LEFT$(A$,1))+128
44 FORK=2TOLEN(A$):J=ASC(MID$(A$,K,1))
45 L=L+1:POKEL,J:NEXT:NEXT
46 L=L+1:POKEL,171:L=L+1:POKEL,173:L=L+1
47 POKEL,170:L=L+1:POKEL,175:L=L+1:POKEL,219
48 L=L+1:POKEL,197:L=L+1:POKEL,83:L=L+1
49 POKEL,214:L=L+1:POKEL,65:L=L+1:POKEL,71
50 L=L+1:POKEL,89:REM ES,VAGY
51 FORI=1TO128:POKEL+I,PEEK(6049+I):NEXT
52 POKE16526,0:POKE16527,100:A=USR(0)
53 PRINT"KESZ! MAGYARITOTT KULCSSZAVAK."
54 END: REM DE LEHETNE NEW
55 DATA VEGE,SORBA,PONTKI,PONTBE,TOROLJ,CMD
56 DATA VELETLEN,KOVETKEZO,ADAT,BE,HELY
57 DATA OLVASS,LEGYEN,MENJ,FUSS,HA,UJRA
58 DATA ALPROGRAM,VISSZA,MEGJEGYZES,ALLJ
59 DATA KULONBEN,TRON,TROFF,STRING,EGESZ
60 DATA EGYSZERES,DUPLA,SOR,JAVIT
61 DATA HIBA ESETEN,FELOLDAS,KI,ELAGAZ
62 DATA NYIT,FIELD,GET,PUT,ZAR,LOAD
63 DATA MERGE,NEV,KILL,LSET,RSET,SAVE,HIVJ
64 DATA NYOMTATORA,DEF,POKE,IRJ,FOLYTASS
65 DATA LISTA,LLIST,TOROLJ,AUTO,NULLAZZ
66 DATA MAGNOROL,MAGNORA,FELEJTS
67 DATA TAB(,TO,FN,USING,VARPTR,USR,ERL
68 DATA ERR,KARAKTERSOR,INSTR,VILAGIT
69 DATA TIME$,MEMORIA,BILLENTYUROL
70 DATA AKKOR,NEM,LEPJ
```

tett kulcsszavai helyett tetszőleges magyar kulcsszavakat alkalmazhatunk. A függvények D7H—FBH kulcsszavai helyett is beírhatunk magyarított megfelelőket, de az itt közölt változatnál már csak helytakarékosság miatt is eltekintettünk ettől.

A program a végrehajtási és a beírási főág megfelelő címeiről elugrik a mi rutinunkba. Vigyázat! A kulcsszótáblát még az EDIT-állás is igényelné, ez az eredeti ROM-ból újabb 500 bájt átmásolását, ezen belül 50 bájt átírását jelentené. Ez itt elmaradt.

Mit tud akkor ez a program? *(2. lista)*

A BASIC nyelvű program futtatásakor a programunk részére fenntartott területre olyan programrészeket ír és másol át, amelyek a tokenizáláshoz és a listázáshoz szükségesek.

13 Az RST vektor átírásának előkészítése
15—17 Honnan hívta meg az RST rutint a BASIC-értelmező? Beépítve a ROM 33D4H-ról a szükséges rész
18 Ha beolvasási főágról, akkor saját tokenizáló rutinra
Ha LIST, LLIST (ide kellene az EDIT is), akkor saját rutinra, egyébként vissza a hívóhoz
23—25 A 2B29H-n kezdődő terület (LIST) átmásolása és a szükséges átírások (kulcsszótábla címe) elvégzése
26—30 Beolvasás, tokenizálás 1A82H, 1BC0H, 1A9CH kezdőcímekről áttöltés, az ugróutasítások címeinek átírása
30—40 Saját kulcsszótábla felépítése az eredeti ROM logikája és sorrendje alapján
42 RST vektor átírása

A futtatással „a mór megtette kötelességét, a mór mehet", azaz ez a program törölhető, de ha most paranccsal akarjuk ezt elvégezni, akkor a NEW helyett FELEJTS kulcsszót kell beírni. Ha a program listáját még egyszer látni akarjuk, akkor LIST helyett LISTA írandó be. Programjainkat most már magyar kulcsszavakkal írhatjuk, futtathatjuk (RUN helyett FUSS).

A magnóról beolvasott, korábban eredeti BASIC nyelven megírt programjaink listája magyar kulcsszavakkal jelenik meg a képernyőn. A mi „magyar BASIC-ül" megírt programunk alapállapotú gépbe visszatöltve a HT eredeti kulcsszavaival jelenik meg. A „magyarul" beírt program listája a bővítés (HIVJ/12288) meghívása után eredeti kulcsszóval jelenik meg a képernyőn. A bővítés meghívása is átírja az RST 10 vektort. Ebben a formájában a magyarított változat visszaállítására ilyenkor nincs lehetőség.

3. lista

```
10 PRINT"  MAGYARITOTT C=16 BASIC "
11 PRINTCHR$(27)CHR$(84):PRINT"BETOLTES"
12 FORI= 818TO 972 :    GOSUB26:NEXT
13 FORI=1630TO1713 :    GOSUB26:NEXT:K=32000
14 FORI=128TO169:READA$:GOSUB22:NEXT
15 FORI=170TO174:READA$:GOSUB24:NEXT
16 FORI=175TO176:READA$:GOSUB22:NEXT
17 FORI=177TO179:READA$:GOSUB24:NEXT
18 FORI=180TO253:READA$:GOSUB22:NEXT
19 POKE780,70:POKE781,3:POKE774,97:POKE775,6
20 POKE774,97:POKE775,6:POKE772,50:POKE773,3
21 NEW
22 FORL=1TOLEN(A$)-1
23 POKEK,ASC(MID$(A$,L,1)):K=K+1:NEXT
24 POKEK,ASC(RIGHT$(A$,1))+128:K=K+1
25 PRINT"*":RETURN
26 READH$:POKEI,DEC(H$):RETURN
27 DATA A5,3B,48,A5,3C,48,20,79,04,4C,41,03,20,73
28 DATA 04,90,FB,6C,0C,03,90,6F,C9,00,F0,55,C9,3A
29 DATA F0,EE,C9,3F,D0,04,A9,99,D0,2E,C9,80,90,0B
30 DATA C9,FF,F0,DE,A0, 1,20,EA,89,F0,D1,C9,22,D0
31 DATA 0D,20,73, 4,C9, 0,F0,2F,C9,22,F0,C8,D0,F3
32 DATA 20,B0, 3,90,C1,C0,00,F0,03,20,EA,89,A5,0B
33 DATA A0,00,91,3B,C9,8F,F0, D,C9,83,D0,AC,20,73
34 DATA 04,20,B0,8D,4C,33, 3,20,73, 4,20, B,8E,A6
35 DATA 3B,68,85,3C,68,85,3B,38,8A,E5,3B,A8,C8,60
36 DATA A9,7D,A0,00,4C,07,8A,48,88,88,20,EA,89,A0
37 DATA 00,A9,FE,91,3B,C8,68,91,3B,20,73,04,4C,3E
38 DATA 03,4C,4F,8B,10,FB,C9,FF,F0,F7,24,0F,30,F3
39 DATA C9,FE,D0,17,C8,20,D1, 4,F0, C,84,49,38,6C
40 DATA 0E, 3,B0,E1,A0,00,F0,24,88,A9,FE,D0,D9
41 DATA AA,84,49,A0,7D,84,23,A0,00,84,22,A0,00,CA
42 DATA 10,0F,B1,22,48,E6,22,D0, 2,E6,23,68,10,F4
43 DATA 30,EF,C8,B1,22,30, 6,20,B2,90,D0,F6,60,4C
44 DATA 4B,8B
45 DATA VEGE,SORBA,KOVETKEZO,ADAT,BE#,BE,TOMB
46 DATA OLVASD,LEGYEN,MENJ,FUSS,HA,UJRA
47 DATA ALPROGRAM,VISSZA,MEGJEGYZES,ALLJ,ELAGAZAS
48 DATA VARJ,HOZZD BE,VIDD KI,ELLENORIZZ,DEF
49 DATA POKE,IRJ#,IRJ,FOLYTASD,LISTA,NULLAZZ,CMD
50 DATA HIVJAD,NYISD,ZARD,HOZZAD,UJ,TAB(,TO,FN
51 DATA SPC(,AKKOR,NEM,LEPJ,+,-,*,/,↑,ES,VAGY,)
52 DATA =,<,ELOJEL,EGESZRESZ,ABSZOLUT,USR,SZABAD
53 DATA POS,GYOK,VELETLEN,LOG,EXP,COS,SIN,TAN,ATN
54 DATA PEEK,HOSSZ,KARAKTERRE,SZAMMA,ASC,KARAKTER
55 DATA BAL$,JOBB$,KOZEP$,GO,ROR,RCLR,RLUM,JOY
56 DATA RDOT,DEC,HEX$,HIBA#,INSTR,KULONBEN,FELOLD
57 DATA HIBAELFOGO,NYOMBE,NYOMKI,HANG,HANGERO
58 DATA AUTO,PUDEF,RAJZ,FEST,HELYEZD,KOCKA,KOR
59 DATA GSHAPE,SSHAPE,PONT,POZICIO,SZIN,TISZTITS
60 DATA LEPTEK,HELP,ELEJE,HUROK,LEPJ KI,DIR,DSAVE
61 DATA DLOAD,HEADER,SCRATCH,COLECT,COPY,RENAME
62 DATA BACKUP,SORTORLES,ATSZAMOZ,BILLENTYUROL
63 DATA MONITOR,USING,AMIG IGAZ,AMIG HAMIS
```

```
10 IRJ"  MAGYARITOTT C=16 BASIC "
11 IRJCHR$(27)CHR$(84):IRJ"BETOLTES"
12 SORBA1= 818TOL 972 :    ALPROGRAM26 KOVETKEZO I
13 SORBA1=1630TOL1713 :    ALPROGRAM26 KOVETKEZO I:K=32000
14 SORBA1=128TOL169:OLVASDA$:ALPROGRAM22 KOVETKEZO I
15 SORBA1=170TOL174:OLVASDA$:ALPROGRAM24 KOVETKEZO I
16 SORBA1=175TOL176:OLVASDA$:ALPROGRAM22 KOVETKEZO I
17 SORBA1=177TOL179:OLVASDA$:ALPROGRAM24 KOVETKEZO I
18 SORBA1=180TOL253:OLVASDA$:ALPROGRAM22 KOVETKEZO I
19 POKE780,70:POKE781,3:POKE774,97:POKE775,6
20 POKE774,97:POKE775,6:POKE772,50:POKE773,3
21 UJ
22 SORBAL=1TOL HOSSZ(A$)-1
23 POKEI,ASC(KOZEP$(A$,L,1)):K=K+1:KOVETKEZO
24 POKEI,ASC(JOBB$(A$,1))+128:K=K+1
25 IRJ"*":VISSZA
26 OLVASDH$:POKEI,DEC(H$):VISSZA
27 ADAT A5,3B,48,A5,3C,48,20,79,04,4C,41,03,20,73,
```

4. lista

Ha már a magyarításnál tartunk, akkor jó lenne ezt a programot más gépeknél is alkalmazni. A primósok figyelmét ezért érdemes felhívni arra, hogy a HT és Primo gépek ROM-jában azonos rutinok találhatók (az első 1228 bájtból 9826 bájton). A két gépet összehasonlítva azonos címen, azonos tartalmú bájtokat találtunk, többek között az

1A22—1A76
1A78—1D1E
1D25—1DA0
1DA2—2072
280C—2B46
2B4A—2B70
2B73—2BF4

területeken. A 48 és 64-es változatok memóriaterületének lehatárolásával csaknem változtatás nélkül beírhatjuk a HT programot a Primóba.

C16, Plus/4

A *3. lista* a bevezetőben elmondottakat teljesen kielégíti. A DATA sorokban csak az alapfokú programozási ismereteken kívül felhasznált kulcsszavakat „magyarítottuk", mert aki már lemezműveleteket végez, az eleve inkább azokat az utasításokat tanulja meg. Aki viszont már haladó, az fejlessze tovább ezeket a programokat úgy, hogy az egészen kezdők hamarabb és értelemszerűen tudjanak megismerkedni a programozás rejtelmeivel. Ha már értik a programozás elemeit, akkor a bőbeszédű magyar kulcsszavak helyett — addig is, amíg nincs jobb — célszerű az eredeti BASIC-változat kulcsszavait megtanulni. A fordítást megkönnyíti, ha a magyarul írt programunkat betöltjük egy nem manipulált gépbe, mert az ott az eredeti kulcsszavakkal olvasható — és viszont (*4. lista*).

SZ. LUKÁCS JÁNOS

3–MEG–1-gyel kényelmesebb . . .

Azok, akik a Commodore Plus/4 beépített programjait használják, méltán bosszankodnak azon, hogy az ékezetes karakteretek a 3—Plus—1 egyetlen programja sem fogadja el. Ezt a problémát igyekszik megoldani az a — külön is megvásárolható — program, amelyet a nyomtatókban levő EPROM kicserélésekor kap meg a felhasználó. Sajnos ez a beszúrás sem tökéletes, mert a szövegszerkesztőben használhatók ugyan az ékezetes karakterek, de a kész dokumentumot nem lehet kinyomtatni. A megpiszkált program a <•p> és a <pr> nyomtatási parancsra vagy nem reagál, vagy lemerevedik. A vele való munka egyébként is elég hosszadalmas: a kész szöveget előbb lemezre kell mentenünk, aztán a gépet kikapcsoljuk, a belső 3—Plus—1-et elinditjuk, a kimentett szöveget betöltjük, és végül kinyomtatjuk. Ha a szöveg hibátlan, akkor szerencsénk van, ugyanis a javításhoz ismét az ékezetes szerkesztőt használhatjuk.

Az alábbiakban egy olyan megoldást adok közre, amely ezt a kényelmetlenséget kiküszöböli, sőt azoknak is segit, akiknek nyugati márkájú nyomtatójuk van, német vagy más idegen nyelvű ékezetes karakterkészlettel.

Ut	Eredeti	Funkcio
(ESC)	(C>)c	Parancs módba lépés
(Shift+Q	(C>)q	Előző parancs ismétlése
(Ctrl)+E	(C>)•E	Hibás (RET) semlegesítése
(F1)	(C>)•r	• (Shift)+(Return)
(F2)	(C>)•l	Széles kurzor balra kurzor középre (41.poz)
		Széles kurzor jobbra
(Ctrl)+x	(C>)•t	Text mód
(Ctrl)+n	(C>)•n	Numerikus mód
(Ctrl)+f	(C>)•f	Formula mód

A módosított program kezelése

Mivel az ékezetes karaktereket a <C= > billentyűvel együtt kell irni, a módositás megkövetelte a szerkesztés közben használandó vezérlők kicserélését. Arra törekedtem, hogy az új vezérlők lehetőség szerint könnyen memorizálhatók legyenek, és a <C= > váltó felszabaduljon.

A módosítások lényege

A változtatások két csoportba sorolhatók. Egyrészt az eredeti vezérlőket új karakterekkel helyettesitettem. Ezzel felszabadultak az ékezetes karakterek. A $178C—$1794 tartományon pedig a rendszer által alkalmazott ASCII-SCREEN kódkonverziót változtattam meg.

Természetesen a nyomtató csak akkor irja az ékezetes karaktereket, ha van ilyen készlete. Ezért a RAM-ba töltendő készletnek meg kell egyeznie a nyomtató saját (magyar szabvány vagy más szabvány szerinti) készletével.

A 3–MEG–1 elkészítése

A módosító program három részből áll. Az első egy rövid BASIC inditó, amely akár el is hagyható. Az inditást a SYS DEC('1400') utasitás kezdeményezi. A gépi kódú program (*1. lista*, illetve ennek assembler formája a *2. lista*) első lépésben a karaktergenerátort átmásolja a $1800—$1FFF területről a $F000—$F7FF zónába. Második lépésben a $0700—$07A0 a szabad rendszerterületre irja be a megszakító rutin elé helyezett beszúrást és átírja a $0314—15 címen az IRQ vektort, melynek eredeti értéke $CE0E. Végül a program a 3—Plus—1 inditórutinjára ($05F5) adja a vezérlést.

Ha a 3—MEG—1 programot olyan gépre alkalmazzuk, amelyben a karakter-ROM az ékezetes készletet tartalmazza, akkor a gépi kódú részből az első blokk és természetesen az egész programból a harmadik részt alkotó karaktergenerátor elhagyható.

A három részt külön elkészitve lemezre mentjük. Innen egyenként a RAM-ba töltjük és átírjuk a $2D—$2C BASIC-vége mutatót $2000-re. Az igy egybefüzött programtörzset normál DSAVE '3—MEG—1' rendszerparanccsal irjuk lemezre. A gépi kódú rész leirásához azok, akiknek nincs assemblerük, a MONITOR-ral készült memóriadumpot használhatják. A nyomtatóéval azonos karakterkészlet elkészitéséhez természetesen karakterszerkesztő programra van szükség.

```
>1400 AD 06 FF 29 EF 8D 06 FF :
>1408 A2 00 BD 00 18 9D 00 F0 :
>1410 E8 D0 F7 EE 0C 14 EE 0F :
>1418 14 AD 0C 14 C9 20 D0 EA :
>1420 A9 C0 8D 12 FF A9 F5 8D :
>1428 13 FF A2 00 BD 46 14 9D :
>1430 00 07 E8 E0 A0 D0 F5 78 :
>1438 A9 00 8D 14 03 A9 07 8D :
>1440 15 03 58 4C F5 05 A9 00 :
>1448 8D 47 05 A9 18 8D 04 11 :
>1450 8D E7 19 A9 22 8D EC 10 :
>1458 8D 3D 13 A9 1C 8D 40 13 :
>1460 A9 86 8D 08 11 8D 3E 13 :
>1468 8D 6A 32 8D 86 38 A9 85 :
>1470 8D 8C 11 8D 3F 13 8D 65 :
>1478 32 8D 8E 38 A9 18 8D 18 :
>1480 11 A9 06 8D 1C 11 A9 0E :
>1488 8D 28 11 A9 10 8D 8C 17 :
>1490 A9 07 8D 8D 17 A9 09 8D :
>1498 8E 17 A9 40 8D BF 17 A9 :
>14A0 EA 8D 93 17 8D 94 17 AD :
>14A8 86 7D D0 30 A9 AA 8D 43 :
>14B0 51 A9 C0 8D 4A 52 A9 22 :
>14B8 8D 88 55 A9 86 8D AF 68 :
>14C8 A9 85 8D B9 68 A9 18 8D :
>14D8 CC 68 A9 06 8D D5 68 A9 :
>14E0 8E 8D C3 68 A9 1B 8D DE :
>14E8 68 4C 0E CE A9 1B 8D B2 :
>14F0 68 4C 0E CE 00 00 00 00 :
>14F8 00 00 00 00 00 00 00 00 :
>1500 00 00 00 00 00 00 00 00 :
```

1. lista

A beszúrás működése

Az assembler listában látható IRQ beszúrás végzi el a beépített felhasználói programokban a módosításokat, majd a KERNAL megszakitási rutinra ($CE0E) adja a vezérlést. Ez a módosító a már emlitett ékezetes beszúrástól nemcsak anyniyban különbözik, hogy más vezérlőket használ, hanem abban is, hogy a módositások egy részét feltételesen végzi el. Ehhez a $7D06 cimen levő állapotjelzőt kell megvizsgálni. A beépitett rendszer ugyanis nem fér el a szabad RAM ($1000—$7FFF) területén. Ezért a program készitői ún. lapozásos technikát alkalmaztak: bizonyos funkciók kiszolgálásakor a $4A00-tól kezdődő zónát a ROM-ból az aktuális modullal töltik fel. Az érvényes modul lapozókódját a $7D06 jelző tartalmazza.

DR. HACK FRIGYES

2. lista

```
:put"00:ass-3+1"
itt     = $1800         ; -2000 set 1-2
ram     = $f000
flag    = $7d06
        * = $1400               irq
        lda $ff06       ; video ki
        and #$ef
        sta $ff06
        ldx #0          ;  ram ← set 1-2
cikl    lda itt,x
        sta ram,x
        inx
        bne cikl
        inc cikl+2
        inc cikl+5
        lda cikl+2
        cmp #$20
        bne cikl
        lda #$c0
        sta $ff12       ; ted12
        lda #$f5
        sta $ff13       ; ted13
;
        ldx #0          ; $700 ← irq
irl     lda irq,x
        sta irq,x
        inx
        cpx #$a0
        bne irl
        sei             ; irq vektor
        lda #$00
        sta $314
        lda #$07
        sta $315
        cli
        jmp $5f5        ; 3-plus-1
irq     lda #0
        sta $547        ; set 1<->2
        lda #$1b        ; <esc> bc=c
        sta $1104
        sta $19e7
        lda #$22        ;  <"> ab=q
        sta $10ec
        sta $133d
        lda #$1c        ; ctr+4  a4=@
        sta $1340
        lda #134        ;  <f2>  b2=r
        sta $1108
        sta $133e
        sta $326a
        sta $3886
        lda #133        ;  <f1>  b6=l
        sta $110c
        sta $133f
        sta $3265
        sta $388e
        lda #$18        ; ctr+x  a3=t
        sta $1118
        lda #$06        ; ctr+f  bb=f
        sta $111c
        lda #$0e        ; ctr+n  aa=n
        sta $1120                       nem
        lda #$10        ; konverzio
        sta $178c
        lda #$07
        sta $178d
        lda #$89
        sta $178e
        lda #$40
        sta $178f
        lda #$ea
        sta $1793
        sta $1794
        lda flag
        bne nem
        lda #$aa                ;  !
        sta $5143
        lda #$c0                ;  -
        sta $524a               ;  "
        lda #$22                ;  ab
        sta $5580
        lda #134                ;  b2
        sta $68af
        lda #133                ;  b6
        sta $68b9
        lda #$18                ;  a3
        sta $68cc
        lda #$06                ;  bb
        sta $68d5
        lda #$0e                ;  aa
        sta $68c3
        lda #$1b                ;  bc
        sta $68de
        jmp $ce0e
        lda #$1b                ;  bc
        sta $60b2
        jmp $ce0e
```

2. feladat: Snóbli

Írjon olyan snóbliprogramot, amely kihasználja az ellenfele játékában felismerhető szabályszerűségeket!

A megoldás

Az ilyen program akkor éri el a célját, ha van benne olyan rész, amely számolja az ellenfél játékában felismert sorozatok előfordulását. Ez a *2. listán* látható programban (ZX-Spectrum BASIC) az S(2,256) tömb. Ennek második indexe a sorozat száma, amelyet úgy kapunk, hogy a játékos utolsó nyolc lépését egymás mellé írjuk (például FFIFIFII), majd a fejet 0-val, az írást 1-gyel helyettesítjük. Az index az így kapott bináris számnál eggyel nagyobb. (Azaz példánkban: 00101011B+1=44D.) A tömb S(1,44) eleme azt mutatja, hogy egy ilyen sorozat után hányszor következett fej, S(2,44) pedig azt, hogy hányszor következett írás.

Ha egy ilyen tömbünk van és ismerjük a játékos előző nyolc lépését (S változó), akkor ezt felhasználhatjuk és ennek arányában tippelhetünk (310-es sor).

Minden lépés után a programnak „tanulnia" kell. Ezt az 1000-es sorban található szubrutin végzi. Az 1010-es sor S tömb, az 1030—1040-es sorokban az utolsó nyolc lépést tároló S változó értéke módosul.

```
 TALALT :  1183          SNOBLI
 NEM TALALT:  905
 OSSZESEN :  1988
 TALALATI ARANY:  0.59507042

        *** JATEK ***
 ENYEM:  F I I I F F I F I F
 TIED:     I F F I I F F I I I
   (ENTER): = STATISZTIKA
     HIT TIPPELSZ? (F/I)
```

1. ábra

A program fő paraméterei a 160-as sorban találhatók. Itt határozzuk meg, hogy hány lépés mélységig vizsgáljuk a sorozatokat, és azt, hogy a „tanulás" milyen mértékben befolyásolja a játékot. A program további érdekessége, hogy az <ENTER> lenyomásakor az így készült statisztikát kirajzolja. Az ezt elvégző programrészlet a 900—950-es sorokban látható.

A megoldás többi része a beolvasás és kiíratás adminisztrálásával foglalkozik. Beolvashatunk a Kempston botkormányról is. Ez megkönnyíti nagyszámú (több ezer) játéknál az adatbevitelt. Az *1. ábra* segítségével a kiírást könnyebben megérthetjük. Ugyancsak ez példázza a program hatékonyságát: mindössze 1988 játék (ez meglehetősen kevés) után a gép közel hatvan százalékban biztonsággal nyer.

Sorozatunkat elsősorban középiskolásoknak szánjuk, de reméljük, hogy minden olvasónknak tanulási lehetőséget és szórakozást nyújt.

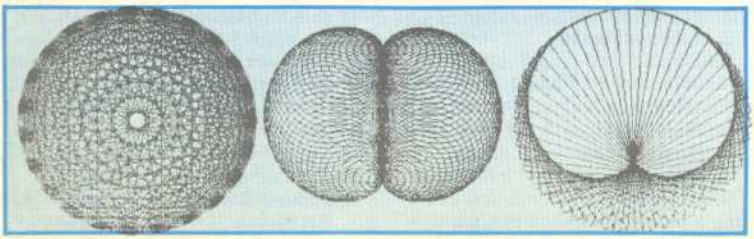
A következő feladat előtt vizsgáljuk meg az *1. listán* látható programot. Ez a *2. ábrán* bemutatott sokszöget összes átlójával együtt rajzolja ki. A program ZX-Spectrum Beta Basicben készült, de csak egy utasításban tér el a standard BASIC-től. Ez az utasítás a 360-as sorban található, és az a feladata, hogy az X(J), Y(J) pontot az X(L), Y(L)-lel összekösse.

```
   10 REM     ---  ATLOK  ---

   20 REM PARAMETEREK BEALLITASA
   30 INPUT "HANY SZOG? ";N
   40 REM     KOZEPPONT
   50 LET X0=127
   60 LET Y0=95
   70 REM     SUGAR
   80 LET R=Y0-1
  100
      REM CSUCSPONT-KOORDINATAK
  110 DIM X(N)
      DIM Y(N)
  200 REM     SZAMITAS
  210 LET D=(PI+PI)/N
      LET S=D
  220 FOR I=1 TO N
        LET X(I)=X0+R*COS S
        LET Y(I)=Y0+R*SIN S
        LET S=S+D
      NEXT I
  300
      REM        RAJZOLAS
  310 LET N2=N/2
  320 FOR I=INT N2 TO 1 STEP -1
  330   FOR K=N
          IF I=N2 THEN LET K=N2
  340   FOR J=1 TO K
  350     LET L=I+J
          IF L>N THEN LET L=L-N
  360     PLOT X(J),Y(J)
          DRAW TO X(L),Y(L)
  370   NEXT J
  380 NEXT I
```

1. lista

2. ábra *3. ábra* *4. ábra*



A program tanulmányozása elsősorban azok számára hasznos, akik nem tanultak még koordinátageometriát és szögfüggvényeket. Ebben a programban ugyanis megtalálható az összes olyan jellegű ismeret, amely a 2. feladat megoldásához szükséges. Három jól elkülöníthető rész fedezhető fel benne:

— A paraméterek beállítása. Ezek olyan állandók, amelyek a számítógép képméretétől és felbontásától függőek.

— A csúcspont-koordináták kiszámítása. Itt X(I), Y(I) egy N-szög I-edik csúcsának

```
   10 REM --- SNOBLI ---
   20 REM  KEP FIX RESZ
   30 GO SUB 710
   99
  100 REM  VALTOZOK DEFINIALASA
  110 LET TALALT=0: LET OSSZES=0
  120 REM KIIRASHOZ HASZNALT V.
  130 DIM E$(2,2): DIM T$(2): DIM
  F$(2,2): LET F$(1)="F": LET F$
  (2)="I"
  140 REM TIPPELESHEZ HASZNALT V.
  150 READ MELYSEG,SULY
  160 DATA 2*5,2: REM PARAMETEREK
  170 DIM S(2,MELYSEG): FOR I=1 T
  O MELYSEG: LET S(1,I)=SULY: LET
  S(2,I)=SULY: NEXT I: LET MAX=SUL
  Y*SULY
  180 RANDOMIZE : LET S=INT ((MEL
  SEG-1)*RND)+1
  199
  200 REM  AZ ALLAS KIIRASA
  210 GO SUB 800
  299
  300 REM  TIPPELES (GEP)
  310 LET ET=INT (S(1,S)/MAX*RND+S(2,S+1
  ))>S(1,S+1)
  320 LET E$=E$(3 TO )+F$(ETIPP+1
  )
  399
  400 REM  TIPPELES (JATEKOS)
  410 BEEP .005,50
  420 POKE 23658,8: REM CAPS LOCK
  430 IF INKEY$="I" OR IN 31=2 TH
  EN LET JTIPP=1: GO TO 470
  440 IF INKEY$="F" OR IN 31=1 TH
  EN LET JTIPP=0: GO TO 470
  450 IF INKEY$<>"" OR IN 31<>3 OR IN 31=
  16 THEN GO SUB 900: GO SUB 700: GO SUB 900
  460 GO TO 420
  470 LET T$=T$(3 TO )+F$(JTIPP+1
  )
  480 LET T=JTIPP: GO SUB 1000
  499
  500 REM  ERTEKELES
  510 LET OSSZES=OSSZES+1: LET TA
  LALT=TALALT+(JTIPP=ETIPP)
  520 GO TO 200
  599
  600 REM  SZUBRUTINOK
  699
  700 REM  KEP FIX RESZ
  710 BORDER 5: PAPER 5: BRIGHT 1
      INK 0: CLS
  720 PRINT AT 1,7;" TALALT :";TALALT;" NEM
  TALALT :";OSSZES-TALALT
  730 PRINT PAPER 1; INK 7;AT 1,2
  ;" ";AT 2,23;" SNOBLI ";
  AT 3,23;"
  740 PRINT AT 13,8; INK 2;"***
  JATEK ***";AT 14,13;"ENYEM:";AT 15,13;"TIED:
  "
  750 RETURN
  799
  800 REM  AZ ALLAS KIIRASA
  810 PRINT AT 13;TALALT;AT 3,1
  3;OSSZES-TALALT;AT 6,13;OSSZES
  820 IF OSSZES THEN PRINT AT 8,1
  5;TALALT/OSSZES
  830 PRINT AT 15,9;E$,AT 17,9;T$
  840 PRINT INK 3;AT 19,4;"(ENTER
  ): = STATISZTIKA";AT 21,6;"HIT TI
  PPELSZ? (F/I)"
  850 RETURN
  899
  900 REM  STATISZTIKA
  910 CLS
  920 LET M1=175/MAX: LET M2=MELY
  SEG*256
  930 FOR I=0 TO 255: LET J=1+INT
  (I*M2): PLOT I,151+J*M1*(2-J): *
  M1: NEXT I
  940 IF INKEY$="" AND (IN 31>=31
  OR IN 31<8) THEN GO TO 940
  950 RETURN
  999
 1000 REM  TANULAS
 1010 LET S(T+1,S+1)=S(T+1,S+1)+1
 1020 IF S(1,S+1)+S(2,S+1)>MAX TH
 EN LET MAX=S(1,S+1)+S(2,S+1)
 1030 LET S=S+S+T
 1040 IF S>=MELYSEG THEN LET S=S-
 MELYSEG
 1050 RETURN
```

2. lista

x és y koordinátáját jelöli.

— A tényleges rajzolás. Minden csúcsot minden más csúccsal össze kell kötni, ami kissé szokatlanul készül el.

3. feladat: Rajzoljunk

Írja meg azt a két programot, amely a *3.* és *4. ábrán* látható képet kirajzolja! (2-2 pont)

PINTÉR GÁBOR

Pattogó C Plus/4-re

Ez a játék egy VIC—20-ra készült régebbi játék továbbfejlesztett változata. Cél az összes fekete alapú fehér kereszt átváltoztatása. Ezt úgy érhetjük el, hogy a fekete ponttal (labdával) nekipattanunk a kereszteknek. Vigyázat! Az átváltozott keresztnek nem szabad újra nekimenni! A program három pályát ismer. További pályák generálásához az utolsó néhány sor nyújt segítséget. Az irányítási lehetőségeket és a pontszámot a program kiírja. Az időt a képernyő alján levő vonal mutatja. Hibánál az I billentyű megnyomásával a játék elölről kezdődik, az N-re pedig leáll.

KOVÁCS RÓBERT

```
10 SCNCLR:COLOR4,5,1
20 PRINT"■■■
```

Örökélet C16-ra

Az e havi POKE-ok ismét Tuba Imre olvasónk gyűjtéséből származnak, aki a kódokon kívül egy ötletet is beküldött: az EXORCIST nevű játék egyes szobáit fekete-fehér tévén nehéz látni, mert az átszínezett részek árnyalata megegyezik az átszínezésre várókéval. A következő utasítással ez kiküszöbölhető:

POKE 1Ø822,113:POKE 1Ø826,113:POKE 11Ø48,113

ONE MAN		7847,Ø	SHOOT IT		10165,48
PANCHO		7458,165	SKYHAWK		4763,234
	joker	7509,165	SLIPPERY SID		7299,48
PANIK		13235,173	SLIRP		13264,48
PETALS OF DOOM		15719,173	SPACE PILOT		4866,165
POGO PETE		4823,165			8473,165
PULSAR		9841,173	SUICIDE RUN		5808,173
		10948,173	SUPER GRAN		5220,173
RAIDER		6325,173			10770,8
	fuel	13405,165		shields	8443,173
REACH SKY		8708,173	TRAIL		11989,189
		8869,173		jumps	12167,165
RED MOON		6983,165			12169,173
ROCKMAN 2.		8721,173	TURBO RACER		8234,169
	idő	8872,Ø			8235,Ø
RUNNER		8991,173	WACKY PAINTER		5777,48
	fuel	8787,173	WATER GRAND PRIX		11589,173
SEA STRIKE		4946,165	ZYLON	víz	7501,Ø
SHARK ATTACK		6183,173		damage	8463,165
		4340,173			

Képernyőtükrözés a ZX-Spectrumra

1. lista

```
TÜKI   LD IX, 16384 ; képmem. címe
       LD HL, 22560 ; képmem vége+32
       LD C, 96
       CALL RUTIN
       LD IX, 22528 ;ATTR mem. címe
       LD HL, 23328 ;ATTR mem. vége+32
       LD C, 12
       CALL RUTIN
       RET
RUTIN  LD DE, 64
CIK    LD B, 32
       OR A
       SBC HL, DE
C1     LD A, (IX+Ø)
       PUSH AF
       LD A, (HL)
       LD (HL+Ø), A ; CSERE
       POP AF
       LD (HL), A
       INC IX ;számlálók
       INC HL ;beállítása
       DJNZ C1
       DEC C
       JRNZ CIK
       RET
```

Játékprogramok írásánál humoros megoldás lehet a képernyő tükrözése, majd törlése. Az itt bemutatott két rutinnal nagyszerűen tükrözhetünk. Az egyik *(1. lista)* a képernyő vízszintes középvonalára, a másik *(2. lista)* a függőleges középvonalra tükröz. Ha a két rutint egymás után használjuk, középpontos tükrözést kapunk.

CSAJTAI KORNÉL

2. lista

```
TÜKII  LD IX, 16384
       LD HL, 16415
       LD C, 192
       LD C, 192
CIK    LD B, 16
C1     LD A, (IX+Ø)
       CALL RUTIN
       PUSH AF
       LD A, (HL)
       CALL RUTIN
       LD (IX+Ø), A
       POP AF
       LD (HL), A
       INC IX
       DEC HL
       DJNZ C1
       CALL SUM
       DEC C
       JRNZ CIK
ATTR   LD IX, 22528
       LD HL, 22559
       LD C, 24
FØ     LD B, 16
F1     LD A, (IX+Ø)
       PUSH AF
       LD A, (HL)
       LD (IX+Ø), A
       POP AF
       LD (HL), A
       INC IX
       DEC HL
       DJNZ F1
       CALL SUM
       DEC C
       JRNZ FØ
       RET
RUTIN  PUSH BC
       LD B, 8
       LD D, Ø
       OR A
R1     RRCA
       RL D
       DJNZ R1
       LD A, B
       POP BC
       RET
SUM    LD DE, 16
       ADD IX, DE
       LD DE, 48
       ADD HL, DE
       RET
```

Külső audio jel kezelése

A C64-ben elhelyezett SID (Sound Interface Device) lehetőséget ad egy külső jel bevitelére is, de ez a jel nem kerül be az oszcillátorokba, hanem csak a szűrőkön keresztül megy a kimenetre.

Így bármilyen külső jelet megszűrhetünk a SID segítségével. Ezenkívül lehetőségünk van még a SID 3 hanggenerátorát is „rákeverni", így egyszerre szólhat a külső jellel. A külső jeleket a MONITOR port AUDIO N (5. láb!) és az 1. láb azaz LUMINANCE segítségével vezethetjük a gépbe.

Ezek után áthalad a SID szűrőin és „felbukkan" a 4. láb audio OUT kivezetésén, de már megszűrt, kevert állapotban. A szűréseket a SID 23 és 24-es regisztereivel végezhetjük, mint normál esetben.

Végül egy jó alkalmazási példát szeretnék említeni: videofilmek „alá" egyszerre írhatunk képet, zenét (igazi zenét), C64-gyel készített zenét, s szövegeket is. Vagy akár feliratozhatunk egy idegen nyelvű filmet.

BARTOS GYULA

Az ENTERPRISE kazettás tárolása

Enterprise számítógépen a programot nem lehet WP-be, illetve a szöveget a program helyébe tölteni. Van mégis egy megoldás, amellyel a programot karakteres formában visszük szalagra:

OPEN≠106:"TAPE:" ACCESS OUTPUT
LIST≠106
CLOSE≠106

Ezután belépünk WP-be és az F1 gombbal betöltjük. Ha WP-ből programot akarunk kazettára menteni, nyomjuk meg az F3-at, majd írjuk be:

TAPE: név majd ENTER

Az így mentett programot visszatölthetjük BASIC-be.

GÖBÖLÖS LÁSZLÓ

Analóg óra Plus/4-en

```
10 VOL8
20 COLOR4,3:COLOR0,3:COLOR1,1
30 REM***** SZAM LAP ****
40 GRAPHIC1,1
50 FORJ=7.84880009TO1.5888001STEP-.1063
60 U=(99*COS(J)+160)
70 O=(-99*SIN(J)+100)
80 P=P+1
90 D=(92*COS(J)+160)
100 G=(-92*SIN(J)+100)
110 DRAW1,U,OTOD,G
120 NEXTJ
130 L=50:F=0
140 REM***** ORA MUTATO *****
150 FORK=7.84880009TO1.5888001STEP-.5216
160 Z=(L*COS(K)+160)
170 S=(-L*SIN(K)+100)
180 DRAW1,160,100TOZ,S:SOUND1,900,2:SOUND1,950,2:SOUND1,1000,2
190 IFF=60THENRUN
200 F=F+1
210 Q=78:C=0
220 REM ***** PERC MUTATO *****
230 FORT=7.84880009TO1.5888001STEP-.1063
240 A=(Q*COS(T)+160)
250 W=(-Q*SIN(T)+100)
260 DRAW1,160,100TOA,W:SOUND1,900,2:SOUND1,950,2
270 IFC=60THEN440
280 C=C+1
290 R=84:B=0
300 REM ***** MASODPERC MUTATO *****
310 FORI=7.83880009TO1.5888001STEP-.1063
320 X=(R*COS(I)+160)
330 Y=(-R*SIN(I)+100)
340 IFB=60THEN420
350 B=B+1
360 DRAW1,160,100TOX,Y:SOUND1,900,2
370 DRAW1,160,100TOZ,S
380 DRAW1,160,100TOA,W
390 FORE=1TO258:NEXTE
400 DRAW0,160,100TOX,Y
410 NEXTI
420 DRAW0,160,100TOA,W
430 NEXTT
440 DRAW0,160,100,Z,S
450 NEXTK
```

Az óra segítségével szemléltetni lehet, hogy hogyan alkalmazhatók a Plus/4 grafikus utasításai. A kezdőknek jó gyakorló program lehet és egyben látványos is. A program a szinusz és a koszinusz függvényt használja a mutatók definiálásához (például a 320—as és 330-as sor). Nagyon ügyeljünk a pontos begépelésre, mert egyébként pontatlan lehet az óránk. Sajnos a nyomtatásnál egy kis hiba csúszott a programba, ezért a 440-es sorba a 100 után szúrjunk be egy TO utasítást.

ZILAHY TIBOR

80 kbájtos memóriamásolás ZX-Spectrumra

Bizonyára többeknek van 80 kbájtos Spectrumuk. A 80 kbájt memória kihasználására mutatok be egy példát. Valószínűleg sokan programoztak már gépi kódban, és keserűen tapasztalták — ha tévesen adtak meg egy adatot vagy rosszul szerkesztették meg programjukat —, hogy elindítása után a gép „elszáll". Az alábbi rutin segítségével megmenthető az elveszettnek hitt program és a fordító is, ha a következőket csináljuk:

A programunk futása előtt elindítjuk ezt a rutint, amely az egyik memóriarészt (32768-tól 65535-ig) átmásolja a másik tárrészbe. Ha a gép a saját programunk működtetése közben „elszáll", akkor gépeljük be: CLEAR 24999:OUT191,... (a másik memóriarész száma). Ezután programunk újra futtatható a fordítóval együtt. A rutin indítása előtt be kell gépelnünk a 25001-es memóriacímre, hogy melyik memóriarészben vagyunk (0 vagy 1).

A rutin RESET gombnál is működik, tehát ha benyomtuk a RESET-et, akkor csak az egyik tárrészt törli ki. Ezután hasonló módon járjunk el, mint a rendes „elszállásnál".

Ezzel az eljárással rengeteg időt takaríthatunk meg. (A gépi kódú listát 32768 alá fordítsuk, ha meg akarjuk változtatni kezdőcímét.) A gépi kódú lista betöltője itt látható.

PUSKÁS LÁSZLÓ

```
10 REM PUSHT SOFT
20 REM
30 FOR F=1 TO 50
40 READ A
50 POKE 24999+F,A
60 NEXT F
70 DATA 62,0,254,0,32,31,62,1,50,...,195
   97,175,50,195,97,33,0,128,126,87,43
   ,1,211,191,122,179,62,0,211,191,35
   124,254,0,32,243,201,175,50,195,97,
   62,1,50,195,97,201,0,0,0,0,0,0,0,0
```

HA EGYEDÜL NEM MEGY, GÉPPEL KÖNNYEBB . . .

Tudod-e, hogy a számítógép — minden előtte alkalmazott géptől eltérően — menynyiben új eszköz?

Abban, hogy az embernek nem a fizikai, hanem a szellemi munkájában segít.

NEKED MÁR SEGÍTETT?

Készíts olyan programokat, amelyekkel saját munkádat könnyited meg! Kellemesebb a tanulás kedvenc géped társaságában.

A határidő folyamatos.

Földrajz, matek, ének? Segíthet a géped!

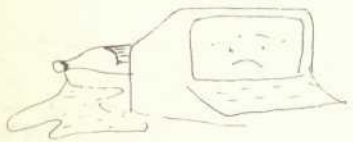
Ha egy háziállattal több van . . .

. . . nem árt néhány információt beszerezni ennek a legújabbnak a természetrajzáról.

Milyen a külseje? Könnyű megkülönböztetni a kabáttól, mert csak gombjai vannak, gomblyukai nincsenek. A jól nevelt számítógép belül hordja a drótkötegeit. Ha mégsem, akkor neveletlen, vagy esetleg drótszőrű foxi.

Rendkívül alkalmas jószág. Hogy mire, azt sejthetjük, mert már több évtizede kutatják. Legnagyobb előnye, hogy nem csíp. Rázni is csak akkor ráz, ha a tenyerünkből próbáljuk etetni.

Szinte az első perctől kezdve szobatiszta. Ha ennek az ellenkezőjét tapasztalnánk, akkor az inkább azt jelenti, hogy kiborult egy üveg kóla.



Napokig kibírja sétáltatás nélkül, és amikor járunk vele egyet, nagyon fegyelmezett. Nem szalad el, és csak rendkívül nagy mennyiségű sör elfogyasztása után — hmmm — közelíti meg túlságosan a sarki villanyoszlopot. Nem kötelező szájkosarat húzni rá. Több nyelve is lehet, de bármennyire elfárad is, egyiket sem lógatja illetlenül. Idomíthatóság szempontjából ideális. Képzett programozók pár heti gyakorlás után percenként akár kétszázszor is kiírathatják vele a képernyőre, hogy „Imádlak Móni" vagy valami hasonló fontosat.

Végül néhány gyakorlati tanács a korszerű számítógéptartás és -gondozás témaköréből.

Az első héten néha emeljük fel az ujjainkat a billentyűkről, hogy láthatatlan pórusain keresztül levegőhöz jusson (bőrlélegző!).

A második héten vessünk néhány oldalpillantást az aggódó őseink által körben felhalmozott táplálékokra, ha a „kidobom azt a vacakot" felkiáltásaikra nem is reagálunk különösebben.

Azért a harmadik héten is kapcsoljuk be néha, hogy áramköreit átmossuk egy kissé.

Körülbelül a tizedik héttől kezdve átlagosan kéthetenként töröljük le a port a tetejéről, de a botkormány tövébe települt pókcsaládot ne zaklassuk túlságosan!

SZÁM LÁSZLÓ
alias L. NUMBER

Még egyszer!

Lapunk előző számában a *„Feladatok — megoldások"* listái nyomdatechnikai okokból egyes példányokban rosszul olvashatók. Ezeket újból közöljük.

```
 5 REM ONMAGAT KIIRO PROGRAM
10 READ AS: PRINT AS:
20 FOR I=1 TO LEN AS
30 IF A$(I)="""" THEN PRINT """
40 PRINT AS(I);
50 NEXT I: PRINT """"
60 DATA" 5 REM ONMAGAT KIIR
0 PROGRAM                    10 READ AS: PRINT A
5:                             20 FOR I=1 TO LEN A
5                                30 IF A$(I)="""""
  THEN PRINT """""                      40 PRINT AS
[1];                                     50 NEXT I
PRINT """""                              60 DATA
```

1. ábra

```
PROGRAM ONKIIRO;
VAR
I,J: INTEGER;
A:ARRAY[1..23]
OF PACKED ARRAY
[1..17] OF CHAR;
BEGIN
A[1] := PROGRAM ONKIIRO;
A[2] := VAR
A[3] := I,J:INTEGER;
A[4] := A: ARRAY[1..23]
A[5] := OF PACKED ARRAY
A[6] := [1..17] OF CHAR;
A[7] := BEGIN
A[8] := FOR I:=1 TO 7 DO
A[9] := WRITELN(A[I]);
A[10] := FOR I:=1 TO 23 DO
A[11] := BEGIN WRITE(A[
A[12] := I,2,'|':=');
A[13] := FOR J:=1 TO 17 DO
A[14] := BEGIN
A[15] := IF A[I][J]=
A[16] := THEN WRITE(
A[17] := WRITE(A[I][J]
A[18] := END;
A[19] := WRITELN(
A[20] := END;
A[21] := FOR I:=8 TO 23 DO
A[22] := WRITELN(A(I))
A[23] := END
FOR I:=1 TO 7 DO
WRITELN(A[I]);
FOR I:=1 TO 23 DO
BEGIN WRITE(A[
I,2,'|':=');
FOR J:=1 TO 17 DO
BEGIN
IF A[I][J]=
THEN WRITE(
WRITE(A[I][J]
END;
WRITELN(
END;
FOR I:=8 TO 23 DO
WRITELN(A(I))
END
```

2. ábra

Egy nap – sok gép

A csákvári Kossuth Művelődési Ház és a TIT Fejér megyei szervezete közművelődési rendezvénysorozat keretében számítógépes találkozót szervez. Mindenkit várnak, aki C64-es, C16-os, C Plus/4-es, Sinclair Spectrum és Videoton TV-Computer típusú géppel dolgozik vagy ilyen gépekkel szeretne megismerkedni. A találkozón profi programozók is részt vesznek.

A látogatók közül a legfiatalabbak tudásukról játékos próbával győződhetnek meg. A serdülő korú gyerekeknek játékversenyt szerveznek, a felnőtteknek pedig bemutatják a számítógép munkahelyi alkalmazásának lehetőségeit. Lesz programcserebere is.

Kezdő számítógépbarátok részére a helyszínen — hozott lemezre vagy kazettára — ingyen úgynevezett Starer Packot (C16 és C64 gépekre) másolnak. A személyi számítógépeken kívül, csemegeként az IBM kompatibilis XT és AT gépek működésébe is belekóstolhatnak az érdeklődők.

A SZÜV Fejér megyei vállalata számviteli és vállalati vezetők részére külön bemutatót is tart ezeken a gépeken.

A találkozót a csákvári sportcsarnokban rendezik meg, 1988. augusztus 20-án. A program és a cserebere reggel 10 órától délután 16 óráig tart. A rendezvényről Honti József (Csákvár, Május 1. út 11. 8083. Telefon: 24/44-307) ad részletes felvilágosítást.

BASIC és gépi kód

Legutóbb néhány FN függvényt és a velük azonos feladatot megoldó, USR-rel hívható gépi kódú rutin C64-re és C16-ra írt változatát ismertettem. Most – a VC20-tulajdonosok türelmét kérve – más témára térek át.

A decimális aritmetika

Az 1987/5. számban, a SED és CLD utasítások ismertetésekor megígértem, hogy később részletesebben is írok a decimális aritmetikáról. Erről a témáról magyarul alig olvasható valami, a nyomtatásban közölt példaprogramok pedig pusztán bemutató jellegűek. Ezért úgy gondolom, hogy mostani írásom érdeklődésre tarthat számot.

A decimális aritmetika csak a két, szűkebb értelemben vett aritmetikai utasításra, az ADC-re és az SBC-re vonatkozik. Minden más utasítás ugyanúgy hajtódik végre, mint bináris aritmetika esetén. A decimális mód használatához meg kell ismerkednünk – egy számunkra új, a számítástechnikában régóta használatos adattípussal – a binárisan kódolt decimális számokkal. Ezeket angol nevük (Binary Coded Decimal) rövidítésével BCD számoknak nevezzük.

A BCD számok egy bájton két decimális számjegyet tartalmaznak, négy-négy biten. Egy BCD számot tartalmazó bájt hexadecimális alakjában csak számjegyet tartalmaz, betűt nem. Általában előjel nélküli egész számok ábrázolhatók BCD alakban, de programozással megoldható az előjelek és tizedesjegyek kezelése is. Az alkalmazható számok jegyeinek száma tetszőleges, csak a memória mérete és az ésszerűség szab határt.

Az ADC és SBC műveletek a bináris aritmetikához hasonlóan az operandus és az A regiszter között mennek végbe, az eredmény az A regiszterben képződik. Ha összeadásnál az alsó félbájton keletkező érték meghaladja a 10-et, a felső félbájt tartalma eggyel nagyobb lesz, és az alsón csak a tízen felüli rész marad. A felső félbájton képződő átvitelt a C bit jelzi. A Z, N és V bitek állapota is változhat, de ezek decimális aritmetika esetén nem nyújtanak használható információt. Kivonásnál hasonló a helyzet.

A műveletek elvégzésekor a processzor nem ellenőrzi, hogy az operandusok valóban érvényes BCD számokat tartalmaznak-e, mindig elvégzi a megadott műveletet. Hibás operandus esetén természetesen az eredmény is hibás lesz. De erről majd máskor írok részletesebben.

1. lista

```
033c 78          sei
033d f8          sed
033e 18          clc
033f ad 60 03    lda $0360
0342 6d 61 03    adc $0361
0345 8d 62 03    sta $0362
0348 d8          cld
0349 58          cli
034a ad 62 03    lda $0362
034d 4a          lsr
034e 4a          lsr
034f 4a          lsr
0350 4a          lsr
0351 09 30       ora #$30
0353 20 d2 ff    jsr $ffd2
0356 ad 62 03    lda $0362
0359 29 0f       and #$0f
035b 09 30       ora #$30
035d 4c d2 ff    jmp $ffd2
```

„Alkotói válságban"

Azzal az elhatározással mentem el a Mikro '88 kiállítás keretében megrendezett újságíró-olvasó találkozóra, hogy azon csak olvasóként veszek részt. Egy – a jelen cikksorozatot is érintő – hozzászólás azonban álláspontom megváltoztatására késztetett, gondolkodóba ejtett, mondhatnám, alkotói válságba sodort. Több, mint két éve fut ez a sorozat, s lehetséges, hogy hiába, mert használhatatlan, akár a BASIC zsebkönyv. (Vagy még annál is kevésbé, hiszen az utóbbi a Delta Impulzus 1988/7. számában közölt sikerlista élén állt, mint a márciusban legkelendőbb műszaki könyv.)

Szeretném, ha sorozatom olvasói megírnák az ezzel kapcsolatos véleményüket. Miről szeretnének e témakörben olvasni, helyes-e, hogy a korábban megjelent információkat ismertnek és hozzáférhetőnek feltételezem, vagy rendszeresen megismételjek-e alapvető dolgokat? Géptől független algoritmusokat ismertessek inkább, vagy az egyes géptípusokon – a ROM-rutinok kihasználásával – gyorsabban létrehozható programokat? Építőelemeket, módszereket mutassak be, vagy kész, begépelhető rutinokat?

Kérem, írják meg ezzel kapcsolatos észrevételeiket, egyéb közleményeiktől elkülönítve, lehetőleg külön lapon. Különösen várom a VC20-ason programozók jelentkezését, hogy megtudjam, érdemes-e időt és energiát fordítani az erre a géptípusra adaptált változatok megvalósítására.

A listákról

A mostani listák egyszerre adnak illusztrációt az előbbi két szakaszhoz.

Az *1. listán* egy gépi kódú rutin látható, az eddig megszokott disassemblált alakban. A rutin decimális módban összeadja

2. lista

```
100 start=$033c
110 chrout=$ffd2
120         sei
130         sed
140         clc
150         lda elso
160         adc masod
170         sta eredm
180         cld
190         cli
200         lda eredm
210         lsr
220         lsr
230         lsr
240         lsr
250         ora #$30
260         jsr chrout
270         lda eredm
280         and #$0f
290         ora #$30
300         jmp chrout
310 elso:   byte $37
320 masod:  byte $55
330 eredm:  byte 0
```

két memóriabájt tartalmát, az eredményt elhelyezi egy harmadik bájton, majd kiírja a képernyőre.

A *2. listán* ugyanennek a rutinnak az assembly forráskódja van. Programjaimat egy speciális hárommenetes assemblerrel készítem, mely felismeri és feldolgozza a BASIC utasítások közé írt assembly kódokat. A három menetet FOR ciklussal kell a BASIC programban létrehozni, az ezzel kapcsolatos utasításokat a listából töröltem. A „start=$033c" értékadás az ismertebb assemblerek „*=$033c" utasításának felel meg. (Kérem, írják meg azt is, hogy célszerűnek tartják-e hasonló forráslista közlését, esetleg a HELP assemblerének formájára átírva. Átírás esetén fennáll annak a kockázata, hogy esetleg valamit közben elhibázok.)

A *3. listával* kapcsolatban is az olvasók véleményét kérem. Ez egy egylépéses szimulátorprogram kimenő listája, és minden utasítás végrehajtása előtt kiírja a regiszterek tartalmát, az állapotregiszterét bitenként, továbbá a végrehajtandó utasítást. Szeretném tudni, hogy az ilyen szimulátorlisták közlése közelebb visz-e a leírtak jobb megértéséhez.

A listák tartalmát a következő alkalommal elemzem. Ezzel kapcsolatban is kíváncsi vagyok olvasóim véleményére: azaz helyes-e, hogy a részletes magyarázat később jelenik meg, mint a lista. Nekem vannak érveim mellette is, ellene is. Önök hogyan látják?

BARNA LÁSZLÓ

3. lista

```
     pc  a  x  y   nv-bdizc
   ------------------------
    033c 00 00 00 %00110001  sei
    033d 00 00 00 %00110101  sed
    033e 00 00 00 %00111101  clc
    033f 00 00 00 %00111100  lda 0360
    0342 37 00 00 %00111100  adc 0361
    0345 92 00 00 %11111100  sta 0362
    0348 92 00 00 %11111100  cld
    0349 92 00 00 %11110100  cli
    034a 92 00 00 %11110000  lda 0362
    034d 92 00 00 %11110000  lsr
    034e 49 00 00 %01110000  lsr
    034f 24 00 00 %01110001  lsr
    0350 12 00 00 %01110000  lsr
    0351 09 00 00 %01110000  ora #30
  * 0353 39 00 00 %01110000  jsr ffd2
    0356 39 00 00 %00110000  lda 0362
    0359 92 00 00 %10110000  and #0f
    035b 02 00 00 %00110000  ora #30
  * 035d 32 00 00 %00110000  jmp ffd2
```


ÁTALÁNYDÍJAS JAVÍTÁSI ÉS KARBANTARTÁSI SZERZŐDÉS = ÖRÖK GARANCIA

FOTOELEKTRONIK ALFA GT NOVOTRADE

COMMODORE, ATARI, TVC stb. személyi számítógépek valamint IBM PC/XT/AT professzionális PC számítógépek és perifériák (floppy, printer) garanciális és fizető JAVÍTÁSA, KARBANTARTÁSA.

SZERVIZEINK:

1053 Budapest V., Magyar u. 12–14. T.: 173-551 Tx: 7621
1083 Budapest VIII., Szigony u. 9. T.: 343-153
1191 Budapest XIX., Gábor A. sétány 3.
3100 Salgótarján, Arany János u. 3. T.: 32-14-007
3525 Miskolc, Fazekas u. 1–3. T.: 46-17-011
4034 Debrecen, Holló László u. 14.
5600 Békéscsaba, Bartók Béla u. 37. T.: 66-27-195
6726 Szeged, Székelysor 13. T.: 62-13-377
7400 Kaposvár, Füredi u. 24. T.: 82-16-307
7624 Pécs, Jurisics M. u. 17. T.: 72-11-812
9700 Szombathely, Szalonok u. 31. T.: 94-14-519

1053 Budapest, Henszlmann I. u. 9. Telefon: 174-144 Telex: 22-7621
VEVŐSZOLGÁLAT — FOTOELEKTRONIK — NOVOTRADE — ALFA G.T.
VÉTEL — ELADÁS 1077 Budapest, Dohány u. 16. Tel.: 428-936

A Precomp Plus előfordító program III.

Plus/4, VC—15XX

DO (UNTIL/WHILE) LOOP
(5. lista 1410—2110*)*

A DO—LOOP ciklus két menetben alakítható át. Az elsőben a DO utasítást tartalmazó sorok sorszámai bekerülnek egy táblázatba (DO% (1-500)). Ennek alapján azonosítható majd a hozzá tartozó LOOP.

A második menetben a DO—LOOP szerkezet helyébe egy FOR—NEXT—STEP ciklus generálódik:

```
10 DO      10 FOR do=1 to 2 STEP 6
20 i=i+1   20 i=i+1
30 LOOP    30 NEXT do
```

A módszerben az a trükk, hogy a DO a fordítandó programban nem tokenizálódik, így változóként kerül fordításra. Mindkét esetben a ciklus addig folytatódik, míg az 1 változó értéke túl nem csordul.

A generált szerkezet mintegy hatszor gyorsabban fut, mint az eredeti.

A LOOP utasításnál is az első menetben az utasítást tartalmazó sor száma egy táblázatba kerül (1800—1802). A második menetben a LOOP helyett NEXT DO utasítás generálódik.

Az UNTIL/WHILE feltételek esetében a feltételt IF—THEN szerkezetbe foglaljuk. Az átalakítást a *6. lista* mutatja.

EXIT (2140—2180)

A DO—LOOP ciklusból való kilépést BASIC 2.0-ban a 6. lista mutatja. A ciklusváltozó értékét megnövelve a NEXT DO utasításra kerül a vezérlés, és az utána következő sorban folytatódik a program végrehajtása.

Hibát a DO-hoz tartozó LOOP utasítás hiánya jelent <DO : DO : LOOP>

GRAPHIC (2255—2500)

A GRAPHIC utasításoknál a Plus—Comp paraméterként csak számkonstansokat fogad el. Változó vagy tömb esetében a Precomp hibaüzenetet ad, és a hiba helyére *-ot ir be. Ha az első paraméter után nem terminátor (:), sorvég (CHR$(0)) vagy „ ,paraméter—2" áll, GRAPHIC SYNTAX hibaüzenetet kap.

```
1410 rem--- do 1.menet ---
1420 d0%(d0%)=(s1%+sh%*256)-32767 :rem" * DO sorszama-32767 tablazatba irasa"
1421 goto 1456
1430 :
1435 rem--- do 2.menet ---
1440 i%=1
1441 fe%=0
1450 c%=Peek(l+i%)
1451 i%=i%+1
1452 if c%=32 then goto 1450
1460 :
1470 print#4,chr$(129);chr$(69);chr$(79);chr$(178);chr$(49);
1480 print#4,chr$(164);chr$(50);chr$(169);chr$(49);chr$(58);
1491 rem                                  " * FOR do=1 to 2 STEP 0
1492 :
1495 rem--- until / while ---
1498 if c%=252 or c%=253 then 1530
1500 hu%=1:h$="+ do":gosub 2960
1510 l=l+i%-1
1511 goto2240
1520 :
1530 hu%=1
1540 if c%=253then fe%=1:h$="+ do while"
1550 if c%=252then fe%=2 h$="+ do until"
1560 gosub2960
1570 fe$=""                              :rem" * Feltetel *
1580 c%=Peek(l+i%)
1581 i%=i%+1
1582 if c%=32 then 1580
1590 fe$=fe$+chr$(c%)
1600 if c%<>58 and l+i%-1<cv then 1580
1610 Print#4,chr$(139);fe$;chr$(167);:rem"* IF feltetel THEN
1611 :
1620 for d=1to d0%-1                     :rem" * DO - LOOP sorszam kereses
1621 if d0%(d)+32767>=so and d0%(d)+32767<=so then 1640
1630 next
1640 sz$=str$(d0%(d)+32767)              :rem" * Sorszam
1650 if fe%=2 then 1760
1660 :
1670 s1%=s1%+2:s2%=sh%                   :rem" * U. sor=sor+1
1672 if s1%>255 then s1%=s1%-255:s2%=sh%+1
1680 c1$=str$(s1%+s2%*256)
1690 Print#4,c1$;chr$(0);
1700 Print#4,chr$(10);chr$(10);
1710 s2%=s1%+1
1711 s4%=sh%
1712 if s2%>255then:s3%=s2%-255:s4%=sh%+1
1720 Print#4,chr$(s3%);chr$(s4%);"do";chr$(178);"2 ";chr$(137);sz$;chr$(0);
1730 Print#4,chr$(10);chr$(10);chr$(s1%);chr$(s2%);":";:rem"U. sor:do=2:goto LOO
P"
1740 goto1790
1750 :
1760 Print#4,"do";chr$(178);"2 ";chr$(137);sz$;          :rem"       do=2:goto LOO
P"
1770 :
1780 l=l+i%-1
1781 goto2340
1790 :
1791 rem--- loop 1.menet ---
1800 l0%(d0%)=(s1%+sh%*256)-32767
1801 d0%=d0%+1
1802 goto1830
1810 :
1811 rem--- loop 2.menet ---
1820 i%=1
1821 fe%=0
1830 c%=Peek(l+i%)
1831 i%=i%+1
1832 ifc%=32then goto1830
1840 ifc%=252orc%=253then goto1900:rem"* UNTIL / WHILE"
1850 :
1860 Print#4,chr$(128);chr$(69);chr$(79);chr$(52);:rem" * NEXT do   "
1861 :
1870 iffe%=0then hu%=1:h$="+ loop" :gosub2960
1880 l=l+i%-1
1881 goto2340
1890 :
1891 rem--- until/ while ---
1900 ifc%=253then fe%=1:h$="+ loop while"
1910 ifc%=252then fe%=2:h$="+ loop until"
1920 gosub2960
1930 fe$=""
1940 c%=Peek(l+i%)
```

ELSE (2505—2640)

IF... THEN... ELSE szerkezet átalakítása a 6. listán látható. Az egymásba ágyazás mélysége olyan lehet, hogy a generált utasítások sorszáma ne legyen nagyobb az (IF) SORSM + 10-nél.

GETKEY

(2655—2750)

A GETKEY kétbájtos kulcsszó, így a Plus—Comp direktíva nélkül szintaktikus hibát jelez fordítás közben a GET kulcsszó után. Ezt a megjelölést végzi el ez a programrész.

USING

(2765—2790)

Hasonló történik a USING esetében is.

DATA

(2805—2940)

A DATA utasítás sorszáma szintén egy táblázat eleme lesz. Ha a RESTORE (n) utasítás hivatkozik erre a sorra, a sorhoz tartozó adatelemek számát kiolvassuk a RESTORE (n) utasításnál leírt módon.

RESTORE (n)

(*7. lista*, 3380-3520)

A DATA mutató sorszámra állítását FOR ciklusba ágyazva READ-del oldottam meg. A DO utasításnál használt trükköt alkalmaztam itt is: ER lett a ciklusváltozó és ERR$ az olvasó sztring. RESTORE után annyi DATA elemet olvastatok ki az ER változójú ciklussal, amennyi az előző DATA sorokhoz tartozó adatszám. Így a következő READ pont azt az elemet olvassa be, amelyre a RESTORE (n) mutatott (*8. lista*).

A Precomp csak sorszámot fogad el a RESTORE után. Nem lehet változó, tömb vagy aritmetikai kifejezés a sorszám. Ellenkező esetben hibaüzenetet küld az előfordító.

Továbbfordítás Plus—Comppal

Az előfordító fordítás végén kiírja a Precomp Plus által hibásnak ítélt utasítások

```
1941 i%=i%+1
1942 ifc%=32then 9oto1940
1950 fe$=fe$+chr$(c%)
1960 ifc%<>59and l+i%-1<cvthen 9oto1940
1970 Print#4,chr$(129);fe$;chr$(167);" rem" * IF feltetel THEN ..."
1971 '
1980 '
1990 iffe%=2then 9oto2020
2000 '
2010 s1%=s1%+2
2011 ifs1%>255then s1%=s1%-255:s2%=sh%+1
2020 z1$=str$(s1%+s2%*256)
2030 Print#4,c1$;chr$(0);" rem"* ... (LOOP sor) + 2
2040 Print#4,chr$(16);chr$(16);
2050 s3%=s1%+1
2051 s4%=sh%
2052 ifs3%>255then s2%=s2%-255:s4%=sh%+1
2060 Print#4,chr$(s3%);chr$(s4%);" rem"* uj sor = (LOOP sor)+1
2070 '
2080 Print#4,"do";chr$(179);"2";chr$(0);" rem"* do=2"
2091 iffe%=1then 9oto2100
2090 s1%=s1%+1
2091 s2%=sh%
2100 Print#4,chr$(16);chr$(16);chr$(s1%);chr$(s2%);" rem"* uj sor (LUUP sor)+1
2110 9oto1960
2120 '
2130 '
2140 rem--- exit 2.menet ---
2150 hu%=1
2160 h$="exit"
2161 9osub 2900
2170 c%=Peek(l+i%)
2171 i%=i%+1
2172 ifc%=32then 9oto2170
2190 '
2190 rem" * do = 2 : GOTO ..."
2206 Print#4,chr$(69);chr$(79);chr$(176);chr$(5d);chr$(52);chr$(137);
2210 forci=1todo%-1
2211 ifd0%(ci)+22767(>oradi0%(ci)+22767,=sothen 9oto2220
2220 next
2221 hu%=2
2222 h$="  exit loop nelkul"
2223 9osub2900
2224 Print#4,chr$(42);
2225 9oto2240
2230 sz$=str$(16*ci)+22767
2231 Print#4,sz$;"     rem" * ...LOOP"
2240 l=l+i%-1
2241 9oto2040
2250 '
2255 rem--- 9raPnic ---
2260 9r%=1
2261 hu%=2
2262 i%=1
2262 h4$=h4$+chr$(c%)
2270 '
2280 c%=Peek(l+i%)
2291 i%=i%+1
2290 ifc%=32then 9oto2280
2300 ifc%=156 then h4$=h4$+chr$(c%):9oto1900  rem"* CLR"
2310 ifc%=44 then P1%=49:9oto2390
2320 ifc%<49orc%>52then h$="  9raPnic Parameter 1 ?":9oto2460
2330 p1%=c%
2340 c%=Peek(l+i%)
2341 i%=i%+1
2350 ifl+i%-1>=cvorc%=58then P2%=48:9oto2420
2360 ifc%=32then:9oto2340
2370 ifc%<>44then h$="  9raPnic syntax?":9oto2460
2380 c%=Peek(l+i%)
2381 i%=i%+1
2390 ifc%=32then 9oto2390
2400 ifc%<48orc%>52then h$="  9raPnic Parameter 2 ?":9oto2460
2410 P2%=c%
2420 n4$=h4$+chr$(P1%)+chr$(44)+chr$(P2%)+chr$(58)+chr$(153)
2421 rem"            * GRAPHIC Parameter1 , Parameter2 : PRINT "
2430 hu%=1
2431 h$="< 9raPnic'
2432 9osub2900
2440 9oto1300
2450 '
2455 rem--- 9raPnic Precomp error ---
2460 9osuL2900
2470 n4$=h4$+chr$(42)
2480 iffl%=1then h4$=h4$+chr$(52)
2490 9oto1290
2500 '
2505 rem--- else ---
2510 hu%=1
2511 h$="< else"
2512 9osub2900
2520 '
2530 i%=1
2531 ifel%=0then:el%=1:cl%=cl%+16   rem" * kovetkezo sor = (ELSE sor) + 10
2540 iffl%=landcl%>255then:cl%=cl%-255:ch%=ch%+1
```

```
2550 rc$=str$(cl%+ch%*256)
2560 c%=Peek(l+i%)
2561 i%=i%+1
2562 if c%=32orc%=127then 9oto2560  rem"* SPACE v39* GOTO
2570 l=l+i%-2
2571 if c%>47and c%<59then Pokel,137 9oto2530  rem"* ELSE sorszám =< GOTO sorszám
2580 Pokel,32
2590 l=l-1
2591 if di%=1then h4$="  "+h4$
2600 if th%=0then h4$=h4$+chr$(127)+kc$  rem"* ... GOTO <ELSE sor)+10
2610 Print#4,h4$;chr$(0);
2611 h4$=""
2620 sl%=sl%+1
2621 if sl%>255then sl%=sl%-255:sh%=sh%+1
2630 Print#4,chr$(10);chr$(10);chr$(sl%);chr$(sh%);  rem"* új sor = <ELSE sor)+1
2640 9oto1040
2650
2655 rem--- 9et / 9etkey ---
2660 i%=1
2670 c%=Peek(l+i%)
2671 i%=i%+1
2672 if c%=32then 9oto2670
2680 if c%<49then h4$=h4$+chr$(161)+chr$(c%):9oto2710  rem" nem KEY
2690 h4$=h4$+" :"+chr$(161)+chr$(c%)
2700 ni%=0
2701 h4$=" 9etkey"
2702 9osub2360
2710 c%=Peek(l+i%)
2711 i%=i%+1
2712 if c%=32then 9oto2710
2720 h4$=h4$+chr$(c%)
2730 if c%=58and l+i%-1<>cvthen 9oto1300
2740 if l+i%-1=cvthen l=l+i%-1 :9oto1050
2750 9oto2710
2760
2765 rem--- usin9 ---
2770 hu%=0
2771 h4$=" usin9"
2772 9osub2360
2780 di%=1
2790 9oto1020
2800
2805 rem--- data ---
2810 i%=1
2811 h4$=h4$+chr$(c%)
2812 da%(da%)=so-32767  rem" * sorszám elhelyezése a tablázatban
2813 da%=da%+1          rem" * DATA elem számlálo
2820 c%=Peek(l+i%)
2821 i%=i%+1
2830 h4$=h4$+chr$(c%)
2840 if c%=:4then 9oto2910
2850 if c%=44then dP%=dP%+1 :9oto2820
2860 if c%=58then dP%=dP%+1 :dP%(da%-1)=dP%:l=l+i%-1 :9oto970
2870 if l+i%-1=cvthen l=l+i%-1 :dP%=dP%+1 :dP%(da%-1)=dP%:9oto1040
2880
2890 9oto2820
2900
2905 rem--- data sztringben ---
2910 c%=Peek(l+i%)
2911 i%=i%+1
2920 h4$=h4$+chr$(c%)
2930 if c%=:4then 9oto2970
2940 9oto2910
```

5. lista

számát. A hiba helyét az előfordított programban *-gal jelölte meg.

A szükséges módosítások elvégzése után a Plus—Comp a hibátlannak ítélt programot már fogadni tudja. Itt van a hibaelemzés a BASIC 2.0-ban értelmezett utasításokon. A BASIC 3.5-ös utasítások szintaktikus elemzését csak a Precomp Plus V—1.1 végzi majd el.

A grafikus fordítási mód szükségességét a „GR: Igen" üzenet mutatja.

Amire vigyázni kell:

Sztringek esetében még PRINT-nél is kötelező a sztringeket „macskakörömmel" lezárni.

Ne használjuk IF szerkezetben utasításhatárolásra egynél több (:) karaktert, mert ez direktivakezelési problémákhoz vezethet.

Ne felejtsük el előfordítás előtt a RENUMBER 10,10-et!

Alkalmazási lehetőségek

A Precomp Plus új sort generáló módszere Commodore 64-en is használható olyan programok „szétszedésére", amelyeket összetömörítettek.

Némi segítséget nyújthat azok számára is, akik programot szeretnének analizálni (feltételes utasítások száma, közvetlen vezérlésátadás, szubrutinhivás, ciklusszervezések hatékonysága stb.).

Más feldolgozási algoritmussal (visszalépés engedélyezésével), szövegkörnyezettől független nyelvi struktúrák elemzéséhez is ajánlható ennek az egyszerű átalakítónak a vizsgálata.

A különböző BASIC nyelvjárások transzformációja így már nem is tűnik nehezen megoldható feladatnak. Ha időm engedi, egy Commodore BASIC 2.0/3.5

6. lista

```
10 rem # do [until/while] loop #
20 i=0:do:i=i+1:if i=10 then exit
30 Print:
40 loop
50
60 i=0:do until i>10:i=i+1:loop
70
80 i=0:do while i<=10:i=i+1:loop
90
100 i=0:do:i=i+1:loop until i>10
110
120 i=0:do:i=i+1:loop while i<=10
130
140 rem # 9raPhic #
150 9raPhic 1,1:9raPhic 0,0:9raPhic clr
160
170 rem # else #
180 a=10:if a=10 then Print"!9xz":else Print"hamis"
190 b=20:if b=20 then b=0:else 210
200 a=c%
210
220 rem # 9etkey #
230 Print"billentsut var" 9etkey a$
240
250 rem # usin9 #
260 a=1234.567:Printusin9"#####.####";a
270
```

```
0 Poke129,128:rem>>b
10
20 i=0
21 fordo=1to2step0
22 i=i+1
22 if i=10 then do=2 9oto 40
30 Print:
40 nextdo
50
60 i=0
61 fordo=1to2step0:if i>10thendo=2:9oto 63
62 i=i+1
63 nextdo
70
80 i=0
81 fordo=1to2step0:if i<=10then 83
82 do=2 9oto 83
83
82 i=i+1
83 nextdo
90
100 i=0
101 fordo=1to2step0
102 i=i+1
103 if i>10thendo=2
104 nextdo
110
120 i=0
```

```
121 fordo=1to2step0
122 i=i+1
123 if i<=10then 125
124 do=2
125 nextdo
130
140
150 9raPhic1,1:Print
151 9raPhic0,0:Print
152 9raPhicclr
160
170
180 a=10
181 if a=10 then Print"!9xz" 9oto 190
182  Print"hamis"
190 b=20
191 if b=20 then b=0:9oto 200
192 9oto210
200 a=c%
210
220
230 Print"billentsut van"
231  :9etkey a$
240
250
260 a=1234.567
261  :Printusin9"#####.####";a
270
```

```
2950 .
2955 rem--- uzenetek kiadasa a kepernyore ---
2960 e%(hu%)=e%(hu%)+1          :rem"* 0/1/2  szint szamlalo
2961 ifhu%<hz%then 9oto3050     :rem"* szintvizsgalat
2970 nu$(no%)=str$(so)+" "+h$   :rem"* uzenet elmentese ismetlesre
2971 no%=no%+1
2980 Printleft$(ki$,7+ki%);sPc(k1%);"■"+str$(so)+"%  ";tab(12);h$
2990 Printleft$(ki$,7+ki%-1);" ";
3000 oPen100,4                  :rem"* nyomtatora is
3001 Print#100,chr$(14);so;chr$(15);".sor: ";h$
3002 close100
3010 h$=""
3020 ki%=ki%+1
3030 ifki%=18then:Print"▒▒▒
```

```
3574 Print#4,"*";
3580 goto 3340
3590 :
3595 rem"* RESTORE n => FOR er= 1 to DATA-1 : READ err$ : NEXT :
3600 ifsz<0then:goto3340
3610 Print#4,chr$(129);"er";chr$(178);"1";chr$(164);str$(sz);":";
3620 Print#4,chr$(135);"err$:";chr$(130);":";
3630 goto3340
3635 :
3636 rem--- forditas vege ---
3640 Print#4,chr$(0);chr$(0);chr$(0)
3650 close2
3651 close4
3652 gosub490
3653 Print#15,"s:"+p1$
3660 close15
3670 Print"■";tab(33)"■"bi$"■■■"
3680 :
3690 char1,0,4,"■Hiba:■"+str$(e%(2))
3700 char1,11,4,"■Csere:■"+str$(e%(1))
3710 char1,22,4,"■ 3,5 :■"+str$(e%(0))
3720 if9r%=1then:a$="i9en":9oto3740
3730 a$="nem "
3740 char1,33,4,"■G:■"+a$
3751 :
3752 ::9etkey a$
3760 :
3770 Print"■■■■■■■Pro9ram : ■";Pn$
3780 Print"■■■■■----------------------■"
3790 Print"■■■■■Keri ujra a hibauzeneteket : ■i■ / ■n■■■"
3900 :
3902 ::9etkey v$::
3903 if v$="n"then:9oto3950
3910 j=0
3911 fori=0tono%-1
3912 j=j+1
3920 Printhu$(i)
3930 if j=15then:j=0:::9etkey a$:Print"■■■■■■■
```

Az Ön feje nem adattárolásra való, hanem fontos döntések hozatalára.
Joggal várhatja el, hogy kezében legyen az eszköz, ami munkáját minőségivé teszi, döntését megalapozza.
A számítástechnikában viszont a széles választékból, nem könnyű a legjobb mellett dönteni.

az Ön fejével gondolkodtunk,

amikor létrehoztuk az első, Közép-Európában egyedülálló számítástechnikai szalont.
Meghívtuk a legfontosabb gyártókat és forgalmazókat, hogy a választék együtt legyen áttekinthető, kipróbálható, tanulmányozható, összehasonlítható.
Felkészült szakembereink várják az érdeklődőket, a leendő vásárlókat.
Reméljük, döntésünk új korszakot nyit az Ön mindennapi munkájában.

PC szalon

Budapest XIII., Sallai Imre u. 6.
☎ 315-136, 310-776

Lépjen új korszakba velünk.

NOVOTRADE

KOMPLEX SZÁMÍTÁSTECHNIKAI SZOLGÁLTATÁS!

A Ferroglobus számítástechnikai főosztálya nagy adatfeldolgozási hagyományokkal, jól képzett, nagy tapasztalatú munkatársakkal rendelkező számítóközpontja széles skálájú, komplex számítástechnikai szolgáltatásait ajánlja:

- **kereskedelmi, szakmai, készletgazdálkodási, pénzügyi, számviteli, bér- és munkaügyi stb. rendszerek szervezése, programozása és fejlesztése nagy kapacitású, modern ICL tipusú számítógépen**
- **mágneses adathordozóra történő adat-előkészítés, -rögzítés**
- **számítógépidő bérbeadása**
- **tanácsadás**

FERROGLOBUS

TEK V. számítástechnikai főosztály

Budapest VII., Vörösmarty u. 16.
Tel.: 427-338, 202-415

MD — GUTOR
szünetmentes áramforrás

A biztonságos termelés alapköve!

0,5 kVA—10 kVA

A legfejlettebb technikát alkalmazó, teljesen zárt rendszerű berendezés.
Óvja értékes számítás- és vezérléstechnikai berendezéseit!

Megbízható megoldás a teljesítményingadozás ellen!

A nyugatnémet—magyar kooperációban készülő berendezések magyarországi forgalmazója:

MD — Vállalkozási és Kereskedelmi Kft.
1112 Budapest, Budaörsi út 42.
Telefon: 666-617 · Tlx: 22-7676 md h

EPSON®

Az EPSON RX—80-as típusú mátrixnyomtatót nem azért ajánljuk Önöknek, mert világbajnok volt a tartós nyomtatásban (2904 órát megszakítás nélkül nyomtatott), még csak azért sem, mert csupán három gomb is elegendő a kezeléséhez, vagy csak hozzá kell illesztenie személyi számítógépéhez, és egy megfelelő programmal máris üzemképes lesz, hanem azért:

— mertmegbízható,
— mertnagy teljesítményű (100 karakter/s),
— mertkönnyű (5 kilogramm),
— mert anyomtatófej éstestékszalag-kazetta hosszú élettartamú,
— mert egykártyás felépítésű éskönnyen javítható,
— mert karakterkészlete tartalmazza amagyar ábécét is,
— mert53 vezérlőparanccsal rendelkezik,
— mert96 ASCII karaktert, 32 grafikus karaktert és 11 nemzeti karaktert tartalmaz,
— mert a cég anyomtatópiac 35,5 százalékát uralja,
— mert12 hónapig garanciális,
— mert az EPSON nyomtatómás, mint a többi . . .

Folyamatosan kapható valamennyi Centrum Áruház Műszaki Osztályán!

	Fogy. ár:
EPSON RX—80 mátrixnyomtató	56 400,— Ft
Ékezetes magyar ABC E—PROM BŐVÍTŐ	2 030,— Ft
Centronics PC kábelcsomag	4 730,— Ft
Enterprise—128 kábelcsomag	4 730,— Ft
Commodore soros interface kábel- és softwarecsomag	18 000,— Ft
Commodore 600/700 PIC printer interface chain, kábel- és softwarecsomag	20 800,— Ft
TV computer nyomtatókábelcsomag	1 630,— Ft

Figyelem! Valamennyi kábelcsomag mellé díjtalanul adjuk a Data Becker sorozat Az EPSON nyomtatók című könyvét.

KERESSE ÁRUHÁZAINKBAN AZ EPSON RX—80-as típusú mátrixnyomtató részletes műszaki tájékoztatóját!

Hardver

A mikroprocesszor

A sorozat alapgondolata — azon a régi felismerésen túl, hogy az elektronika és a számítástechnika elválaszthatatlan egymástól — a következő tapasztalatot summázza. A szoftver — a programok — jelentősége egyre nő, de az is tény, hogy az igazán jó (az adott számítógép nyújtotta lehetőségeket maximálisan kihasználó) programok megírásához a programozónak rendelkeznie kell alapfokú áramköri hardverismerettel is. Megerősíti ezt, hogy szaporodik az olyan berendezések, mikroprocesszort alkalmazó rendszerek száma, amelyek programvezérelten működnek. Az ilyen rendszerek tervezőinek és fejlesztőinek is szükségük van integrált hardver- és szoftverismeretekre.

A mikroprocesszor a mikrogép központi egysége. Az eredeti angol elnevezés alapján szokták CPU-nak is mondani. Jellemzői alapvetően meghatározzák a teljes mikrogéprendszer működését. Funkcióit a következőképpen foglalhatjuk össze:

1. A mikrogép minden elemének biztosítja az időzítő- és vezérlőjeleket.
2. Elvégzi az adatok és utasítások tárból tárba mozgatását.
3. Dekódolja és végrehajtja az utasításokat.
4. Irányítja a be/kimeneti egységekkel kapcsolatos adatforgalmat.

A mikroprocesszor valamennyi funkcióját a benne levő logikai áramkörök végzik. Kívülről ezekhez az áramkörökhöz hozzáférni nem lehet. Áramköri szempontból csak a mikroprocesszor kivezetései érdekesek számunkra. Az itt fellépő jelek szintjei és időbeli viselkedése alapján követhetjük ennek a bonyolult rendszernek a működését.

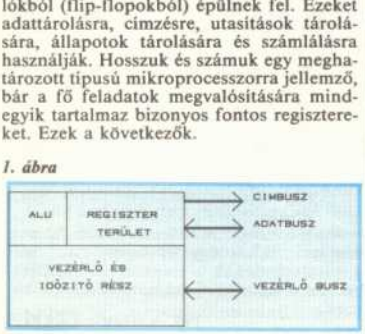
Bár ezek a szekvenciális és kombinációs hálózatokat tartalmazó áramkörök nagyon összetettek, nagyjából mégis három jól elkülöníthető részre bonthatók: regiszterterületre, a vezérlést és időzítést megvalósító területre és az aritmetikai-logikai műveleteket végrehajtó egységre, az ALU-ra. Ezeket az egységeket a hozzájuk rendelhető be/kimenetekkel együtt az *1. ábrán* láthatjuk.

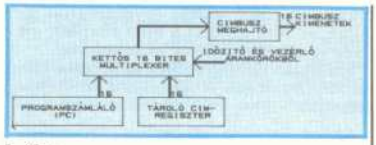
Foglaljuk össze az egyes részekre vonatkozó legfontosabb ismereteket, feltételezve, hogy ismeretesek a gépi kódú programozás alapfogalmai.

A mikroprocesszor regiszterei

A mikroprocesszorban gyakori művelet a bináris információ tárolása, átvitele egyik regiszterből a másikba. A regiszterek tárolókból (flip-flopokból) épülnek fel. Ezeket adattárolásra, címzésre, utasítások tárolására, állapotok tárolására és számlálásra használják. Hosszuk és számuk egy meghatározott típusú mikroprocesszorra jellemző, bár a fő feladatok megvalósítására mindegyik tartalmaz bizonyos fontos regisztereket. Ezek a következők.

1. ábra





2. ábra

Utasításregiszter

Amikor a mikroprocesszor egy utasítást hív le a tárolóból, az utasításnak megfelelő bináris kód az utasításregiszterbe kerül. Az utasításdekódoló egység ezt feldolgozva állítja elő az utasítás tényleges végrehajtásához szükséges logikai vezérlőjeleket.

Programszámláló

A programszámláló mindig annak az utasításnak a címét tartalmazza, amelyet a mikroprocesszornak majd a tárból le kell hívnia.

A tároló címregisztere

Ez a regiszter tárolja annak az adatnak a címét, amelyet a CPU kiolvas vagy beír a memóriába. A címbuszra a címek két forrásból kerülhetnek: a programszámlálóból és a tároló címregiszterből. A kiválasztást a multiplexer végzi, ahogy ezt a *2. ábrán* összefoglaltuk.

Akkumulátor

Az akkumulátorregiszter a legtöbb műveletben részt vevő, kitüntetett szerepű regiszter. Az utasítások nagy része kapcsolódik ehhez a regiszterhez.

Általános segédregiszterek

Ezeket a regisztereket a CPU műveleteihez szükséges adatok átmeneti tárolására használják. Minél nagyobb ezeknek a száma, annál kevesebbszer kell a CPU-nak a külső memóriához fordulnia, így tehát ezek a mikroprocesszorok teljesítményét növelik.

Státuszregiszter

A státuszregiszter bitjei tartalmazzák a CPU által végrehajtott műveletekre vonatkozó információkat. Ezeket flageknek (ejtsd: flegeknek) vagy magyarosan jelzőbiteknek is nevezzük, és mindegyik valamilyen speciális mikroprocesszor-állapotot jelez. A legfontosabbak az átvitelbit, az eredmény negatív volt-át jelző bit stb.

Veremmutató regiszter

Ez a regiszter a veremtárban utoljára tárolt érték címére utal.

A mikroprocesszor időzítő és vezérlő áramkörei

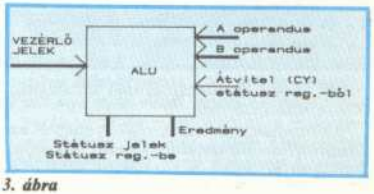
Ezek a jelek teszik lehetővé a CPU és a többi rendszerelem összehangolt működését. Ezek be- és kimeneti jelek lehetnek. Foglaljuk össze a legfontosabbakat.

RESET: bemenet. Mivel a mikroprocesszor egy bonyolult, tárolókat is tartalmazó logikai hálózat, ezért biztosítani kell az alaphelyzetét. Ebbe a helyzetbe a jel aktív szintje állítja a mikroprocesszort.

R/W: kimenet. Ez a vonal jelzi, hogy a CPU olvasási (R) vagy írási (W) műveletet akar végrehajtani. Van olyan mikroprocesszor, ahol külön van írás és olvasási vonal.

MREQ: kimenet. Ez a jel jelzi, hogy folyamatban van a memóriához fordulás.

IORQ: kimenet. Ez a jel jelzi, hogy folyamatban van a be/kimeneti egységhez fordulás.



3. ábra

ALU, a mikroprocesszor aritmetikai logikai egysége

Az adatokkal végzett műveleteket ez az egység hajtja végre, elvi felépítését a *3. ábra* mutatja. A műveletek egy része kétoperandusos, ahol az egyik az akkumulátorregiszterben van, a másik lehet egy adat a memóriából vagy valamelyik belső regiszter tartalma. Az egyoperandusos műveletek legtöbbje az akkumulátorra vonatkozik. A mikroprocesszor utasításkészletének jelentős része kapcsolódik az ALU-hoz. A műveletek eredményére utaló ún. státuszbitek (túlcsordulás, az eredmény nulla stb.) a státuszregiszterbe kerülnek. A mikroprocesszor a memóriában tárolt programot hajtja végre. Kövessük nyomon, hogy hogyan megy végbe egy, a memóriában tárolt utasítás végrehajtása.

A memóriában az adatok és az utasításra jellemző kódok azonos alakban — bináris formában — vannak tárolva. De ha ezek az alakok azonosak, honnan tudja a mikroprocesszor, hogy adott esetben az adat egy utasításkód vagy ténylegesen egy adat? A magyarázat nagyon egyszerű: az értelmezés, illetve a sorrend dönti el. Amikor a mikroprocesszor utasítást hajt végre, feltételezi, hogy a memóriából kiolvasott érték egy utasításkód. A programozónak kell biztosítania, hogy a memóriában az utasítások és adatok sorrendje megfelelő legyen.

Minden memóriához való fordulás egy alapidőtartamot igényel, amit GÉPI CIKLUS-nak nevezünk. A gépi ciklus tovább osztódik kisebb időegységekre, amelyeket ÜTEM-nek nevezünk. Ezekből az ütemekből épül fel az utasítások elvégzése is. Az ütem általában még tovább osztható a rendszer órajelciklusaira. Az órajelek egymásutánjai során alakul ki az utasítás-végrehajtás egy üteme, majd a teljes gépi ciklus.

Egy utasítás végrehajtása mindig az „utasításhozó ciklus"-sal kezdődik. A mikroszámítógép RESET-je után is az első gépi ciklus az utasításhozó ciklus. Az első — a memóriából hozott — adatot tehát mindig utasításnak értelmezi a CPU. A további adatok értelmezését már az utasítás kódja határozza meg.

Az utasításhozó ciklus első ütemében az utasításszámláló tartalma a belső buszmeghajtókon keresztül a címbuszra kerül. Ez eljut a memóriához, az MREQ R/W aktív jelekkel egyetemben kiválasztódik a memória így kijelölt adata és a memória kimeneti regiszterébe íródik. Azt az időt, amíg a memóriánál a cím megjelenésétől az adatkimeneten az adat megjelenik, hozzáférési időnek nevezzük.

A második ütemben az adatvonalon megjelenő adatot a CPU — mivel felismeri, hogy utasítás — a saját utasításregiszterébe juttatja. Ezután a dekódoló azonnal megkezdi az utasítás kódjának megfejtését.

A harmadik ütemben az utasításregiszter dekódolt jelei a vezérlőegységben előkészítik az utasítás végrehajtásához szükséges vezérlőjeleket, és az időzítőegység is megkapja az utasításvégrehajtás időzítéséhez szükséges vezérlőjeleket.

A további — már nem részletezett — ciklusok és ezen belüli ütemek végrehajtásakor képződnek az utasítás operandus(ai) címei, beolvasódnak az azokon levő értékek és ekkor van az utasítás végrehajtása is.

A fentiekben csak röviden vázolt és ténylegesen igen bonyolult működést igazán csak a mikroprocesszor tervezőinek kell ismerniük.

Mintaként bemutatunk egy olyan idődiagramot, amit a mikroprocesszor lábain megjelenő jelek időbeli leírásánál használnak.

Ezeket a digitális jeleket — mivel a H vagy L szintjükkel jellemezhetők — az idődiagramokban egyszerűen csak adott magasságú vízszintes szakaszokkal ábrázoljuk. Az átmeneteket egyik szintről a másikra — utalva a váltás időbeni végességére — ferde vonalakkal jelöljük. Időtengelyt sem alkalmazunk, mivel a jelek időbeliségénél nem a tényleges idő, hanem az egymáshoz való viszonyuk a lényeges. Még két megjegyzés az időbeli ábrázolással kapcsolatban:

— a háromállapotú kimeneteket, mikor azok lebegő állapotban vannak, szaggatott vonallal jelöljük,

— a buszvonalak (cím és adat) jeleit csupán egy párhuzamos egyenespárral jelöljük, és ahol változik a busz állapota, ott ferde vonalakkal jelöljük az átmenetet.

Ezek után vizsgáljuk meg az olvasási ciklus idődiagramját. A jeleket a *4. ábrán* követhetjük.

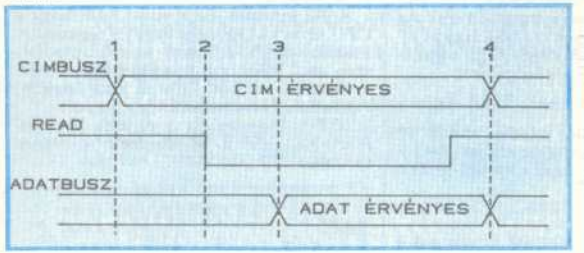
Ahhoz, hogy a processzor valamilyen címről adatot olvasson be, a címkimeneteire kell tennie a címnek megfelelő bináris kódot. Ez a cím a címbuszon megjelenik, és a címbuszra kapcsolódó minden egység bemenetén is megjelenik. Ahhoz azonban, hogy a címen levő adatinformációt a megcímzett egység az adatvonalakra kapcsolja, szükséges még egy jel, amely a tényleges műveletet azonosítja: ez pedig a vezérlőbusz READ (olvasás) jelű vezetékén megjelenő L szintű logikai állapot, ami a processzorba való olvasást jelenti. Ezek után az adat az adatvonalakra kerül és megjelenik a processzor adatvonal-bemenetein. Az ábra felső részén feltüntetett számok a folyamat időbeliségére utalnak: arra például, hogy a cím címbuszra adása és a READ jel megjelenése között idő telik el (1-2 szakasz), vagy pedig arra, hogy a címzett egységnek a 2-4 szakasza áll rendelkezésre ahhoz, hogy az adatot az adatbuszra adja, mert különben a processzor számára elvész.

Egy konkrét mikroprocesszor esetén az áramkört leíró katalógusban ehhez hasonló diagramokkal adják meg a mikroprocesszor különféle működési állapotaihoz tartozó jelek időbeli és egymáshoz viszonyított változását.

Befejezésül a Z80-as lábkiosztását és jeleinek definícióját adjuk meg (lásd az *5. ábrát*).

A mikroprocesszor működésére jellemző idődiagramokat a korlátozott terjedelem miatt itt nem közöljük, de számos irodalomban — magyar nyelven is! — megtalálhatók.

4. ábra



5. ábra

Az 5. ábra jelmagyarázata:

Vcc: a +5 V-os tápfeszültség

GND: a jelek és a tápfeszültség közös testpontja

0: az órajelbemenet

A0—A15: a címsín csatlakozási pontjai. Háromállapotú kimenetek

D0—D7: az adatsín nyolc vezetékének csatlakozási pontjai. Háromállapotú, kétirányú kivezetések

WR: háromállapotú kimenő vezérlőjel, alacsony aktív szinttel, aminek alacsony szintje azt jelzi, hogy a processzor információt küldött ki az adatvonalakra (WRite = írás)

RD: háromállapotú kimenő vezérlőjel, alacsony aktív szinttel, aminek alacsony szintje azt jelzi, hogy a processzor adatot akar olvasni az adatvonalakról (ReaD = olvasás)

MREQ: háromállapotú kimenő vezérlőjel, alacsony aktív szintje azt jelzi, hogy a processzor a memóriával (a tárral) akar adatot cserélni (Memory REQuest = memóriakérés)

IORQ: háromállapotú kimenő vezérlőjel, alacsony aktív szintje azt jelzi, hogy a processzor perifériával (be/kimenetekkel) akar adatot cserélni (Input/Output ReQuest = bemenet/kimenet kérés)

RESET: bemenő vezérlőjel, alacsony aktív szintjének hatására a processzor alapállapotba kerül

M1: kimenet, azt jelzi, hogy utasításlehívási ciklust hajt végre (utasításkódot olvas be) a processzor

WAIT: bemenet, az ide adott alacsony szinttel kell jelezni a processzornak, hogy a megcímzett memória vagy periféria nem kész az adatátvitelre (lassú). A processzor ilyenkor WAIT (várakozó) állapotban marad addig, amíg a bemenet aktív

RFSH: kimenet, aktív alacsony szintje jelzi, hogy a címbusz alsó hét címbitje frissítési címet tartalmaz és a MREQ jel frissítő olvasásra használható. Ennek a jelnek használatát a későbbiekben, a dinamikus memóriáknál értjük meg

Mivel a címbusznak háromállapotú kimenete van, ez lehetővé teszi, hogy nemcsak a processzor képes címet adni a címbuszra, hanem más eszköz is. Ez például lehetővé teszi, hogy a memóriából/ba közvetlenül — tehát nem a processzoron keresztül — olvassunk vagy írjunk. Ilyen módon közvetlen adatforgalmat valósíthatunk meg például a memória és egy mágneslemezegység között. Ezt a megoldást DMA-nak, vagyis magyarul közvetlen memóriahozzáférésnek nevezik. A Z80-nál ezt két vezetékkel alakították ki.

BUSRQ: bemenő vezérlőjel, alacsony aktív szintjének hatására a buszok és a háromállapotú vezérlőkimenetek lebegő (magas impedanciájú) állapotba kerülnek (BUS ReQuest = sínkérés)

BUSAK: kimenő vezérlőjel. Alacsony szintje jelzi, hogy a BUSRQ jel hatására a processzor lebegteti az előzőben említett vezetékeit (BUS AcKnowledge = sínelfogadás)

A processzor fennmaradó három jelének magyarázatához egy új fogalommal kell megismerkednünk, a megszakítás fogalmával. Ezzel következő számunkban foglalkozunk.

DR. KÓNYA LÁSZLÓ

Halálos vírusok?

A legtökéletesebb gyógyítás korában is mindig lesz virussal fertőző bántalom, mely időnként járványszerűen szétterjed az egész földtekén, de legkevesebb annak egy részén végigvonul. A kanyaró, bárányhimlő vagy az influenza rohamosztagainak azonosításához nem szükséges plántált felderítői képesség. Gyorsan, nesztelenül hódít, mint a szerelem. S olyan is: esztelen. De hogyan kerül a csizma az asztalra? Virulencia napjaink egyik legszerelmetesebb, legeszesebb tudományában, a számítástechnikában? Hüledezhetünk, de így van. A tudományos-fantasztikus irodalom rémlátomását, az ősi hitvilág vérszívó lidércnagyát megelevenedni látjuk.

A közelmúlt hónapjaiban az Egyesült Államoktól Nagy-Britanniáig több országban, vállalatoknál és egyetemeken olyan számítógépvírusok fertőzték meg a személyi számítógépekben használatos programokat, amelyek képesek megváltoztatni vagy akár megsemmisíteni bankok, részvénytársaságok, katonai és államigazgatási szervek számítógépes programjait! A számítógépes rendszerekbe becsempészett vírusok tehát roppant kártékonyak lehetnek. Ugyanúgy, mint biológiai kisöcsceik, rendkívül gyorsan terjednek, fertőznek. Egy-egy számítógépes vírus ugyanis képes egyetlen pillanat alatt reprodukálni önmagát, másodpéldányait beágyazni további programokba.

„Alapszabály, hogy amerre terjed az információ, oda vírus is eljuthat" – jelentette ki egy ízben Fred Cohen, a Cincinnati egyetem professzora, aki 1983 óta kutatja a számítógépes rendszereket fenyegető veszedelmet. Mivel a fertőzött programok is fertőzővé válnak, a vírus behatol mindazokba a számítógépekbe, amelyek a „beteg" szoftverrel érintkeznek. A leghevenyebb megbetegedés eddig a személyi számítógépeknél volt észlelhető. Az adatvédelmi és biztonsági szakértők azonban állítják, hogy a legnagyobb veszélyt a stratégiai szempontból domináns nagyszámítógépes rendszerek – mint például a légiforgalmat ellenőrző és irányító, az adóhivatal által üzemeltetett rendszerek – fertőzése jelentené. Kutatások szerint ehhez mindössze egy óra leforgása elegendő. De még egy hatalmas méretű, több világrészre kiterjedő – számítógépek ezreit magába foglaló – nemzetközi hálózatba is be lehet törni néhány nap alatt, azaz ezek sem mentesek a jogellenes betolakodótól. S ami a ragályt még félelmetesebbé teszi az az, hogy vírusfertőzés a számítógépek közötti telefonvonalakon is bekövetkezhet.

Ilyetén úgy tűnik, a szoftverháború, az emberi faj nemzetközi neuraszténiájának újabb neme egyre valószerűbbé válik.

E potenciális tényt megerősíti az, hogy a vírus kijátssza a szokásos ellenőrzéseket és korlátozásokat, amelyeket a felhasználók alkalmaznak információik védelmére. Amerikában egy biztonsági szakember egészen egyszerűen fogalmazott: „A vírus halálos, mert képes átugrani, pontosabban átcsúszni a kiépített védővonalakon. A megoldás: egy erős kapuval ellátott falat kell emelni a dzsungel köré, mivel a számítógépek többségénél jelenleg csak egy álmos kapuőr vigyázza a bejáratot. De ami még ennél is nagyobb gond, azokkal szemben megóvni a rendszert, akik legálisan tartózkodnak, illetve dolgoznak a kapun belül."

Mindazonáltal az USA egyre több államából, egyeteméről jelennek meg tudósítások, amelyek kiterjedt fertőződésről szólnak, és többek között az IBM személyi számítógépek programjait károsítják. De San Francisco körzetében és Nyugat-Európában nagyszámítógépes rendszerek is áldozatul estek már, meghibásodtak. Az egyik legaggasztóbb hír Izraelből származik, ahol az elmúlt év őszén két hónap alatt széles körben elterjedt egy víruskód, melyet látszólag politikai célból, a tiltakozás fegyvereként szándékoztak használni.

Yuval Rakavy, a Héber Egyetem diákja szerint – aki felfedezte, majd hatástalanította a vírust – ez a program időzített bombaként működött volna, amelynek hatására idén május 13-án, pénteki napon a megfertőzött programoknak törölniük kellett volna az összes tárolt adatállományt!!! Május 13-a egyébként a politikai egységként létező Palesztína utolsó napjának 40. évfordulója volt. Izrael 1948. május 14-én deklarálta függetlenségét. Rakavy beszámolójából kiderül, hogy amikor tavaly december 30-adika előtt barátja felkérte, segítsen neki kivizsgálni, miért nem működik jól a személyi számítógépe, hallott már arról: Izraelben terjed a vírus: „Megnéztem a számítógépet és rögtön láttam, miről van szó. Miközben várta a május 13-i időpontot, a vírus már utasította a számítógépet arra, hogy a bekapcsolást követő 30 perc elteltével normál sebességét egyötödére csökkentse és időnként „szemetet" jelezzen a képernyőn."

Végül is az izraeli lidércet a szerencsével karöltve leplezhették le. „Az időzített bomba felfedezése a vírusprogramban lévő hibának volt köszönhető, mert a „merénylő" a korábban fertőződött programokat nem fertőzöttnek tekintette. Ezután tévesen önmaga másolatát hozzáadta a programhoz" – állapította meg Shamuel Peleg, a Héber Egyetem számítógéptudományi karának professzora. Voltak olyan programok, amelyek így 300-400-szor fertőződtek.

A bábeli zűrzavart előidéző akciónál enyhébb szabotázs is elképzelhető. Amikor akár közhuszár vagy hadnagy vág eret alattomban kapitányán, ami rosszabb lehet a kígyómarásnál. – Elméletileg a vírusok felhasználhatók fegyverként a vállalati belharcoknál és kihathatnak a termelésre is. A klasszikus forgatókönyv alapján egy cég alkalmazottja, elnökhelyettese vírussal megfertőzheti az elnöknek a tervkészítéshez használt programjait, hogy annak alkalmatlanságát bizonyítva előnyökhöz jusson, esetleg helyébe léphessen. Ám ugyanígy valamely vállalat versenytársának folyamatvezérlő programját is lesántíthatja.

A szinte bénító hatású számítógépes vírus ellen található-e megfelelő védőrendszer a fentebb említett nagyfalas módszeren kívül? Ez idő tájt a homeopathia, a hasonszervi gyógymód látszik erre a legalkalmasabbnak. Azaz, amikor hasonlót hasonlóval akarnak gyógyítani, tehát egy betegséget oly gyógyszerrel orvosolni, amely egészséges testben hasonló tüneteket idéz elő. Noha az eddig felfedezett vírusok többségükben kártékonyak, elképzelhető jóindulatú, üdvösebb változatuk is. Sok programozót foglalkoztat egy olyan rejtett vírustípus alkalmazásának lehetősége, mely konstruktív célokat szolgálna. San Francisco közelében egy személyi számítógépes szoftverház kifejlesztett egy olyan vírust, amely számolja a generálásokat és nyomonköveti a programduplikációt. Minneapolisban kidolgozták az úgynevezett Data Physician-t, amely azonosítja, hatástalanítja és eltávolítja az IBM PC, valamint a Unix rendszerekből a vírusokat. A cikkünk elején idézett Fred Cohen olyan vírust ajánl, mely mindenkor engedélyt kérne mielőtt akcióba lép és matematikailag sűrítené a kódolást, hogy így az adatok kisebb helyen legyenek tárolhatók.

Dennis Steinaunak, az USA Országos Szabványügyi Hivatala vezető biztonsági szakértőjének véleménye szerint: „Úgy tűnik ez lesz az év játéka, s azt figyeljük majd, hogy ki áll elő a legjobb vírussal." Csak az év játéka? Ez kissé túlzott optimizmus. Terroristák, bűnözők vagy pusztán otromba tréfacsinálók képesek lesznek és egyre képesebbek olyan vírusokat előállítani, melyekkel vandál pusztítást vihetnek véghez. Annál inkább, minél jobban számítógépfüggővé válik egy emberi társadalom. Ez pedig nem más, mint játék kontra fair play. Avagy: „Az ismerethalmaz nem tanít meg az értelemre." – De úgy látszik, hiába mondotta ezt már Herakleitos.

Rajzolóprogram IS-BASIC-ben

Ez a program a gép négyszínű, 720 x 1200 pont felbontású grafikus lapját használja.

A kurzort nyolc irányban mozdíthatjuk el, amit a képernyőn lévő pont azonnal követ, és vonalat húz maga után. Ha a rajzot máshol akarjuk folytatni, akkor az /m/ paranccsal mozgathatjuk el a kurzort, de a felbontást futás közben is megváltoztathatjuk.

A MENÜ PARANCSAI

A / / jelek a közéjük írt alapállapotú — kisbetűs — billentyű lenyomását jelentik.

/f/ — a felbontás megváltoztatása (FELB rutin)

Alapértelmezésben a felbontás 20 pont. Ha kis ívű görbéket rajzolunk, célszerű a felbontást mintegy 3-ra finomítani. A hosszabb vonalszakaszokhoz viszont érdemes növelni. A javasolt intervallum 3—300 között van.

/m/ — mozgatás vonal nélkül (MOZG rutin)

A botkormány mozdítására a érvényes felbontással elmozdul a villogó pont.

/p/ — alakzat betöltése (TOLT rutin)

A kifestő eljárás először a mozgatórutint hívja. Ha a pontot a megfelelő helyre állítjuk (blank), akkor hatására az 1-es paletta színnel kifesti a zárt alakzatot. Ekkor a rutin az IS-BASIC PAINT utasítást használja.

Figyelem! Csak biztosan zárt alakzatot fessünk ki! Egy piciny lyukon is „kifolyhat" a festék, és tönkreteheti addigi munkánkat.

/t/ — szakaszok törlése (TORL rutin)

A program az utolsó ötven törléspont koordinátáját tárolja. A /t/ billentyű egy-egy lenyomására egy-egy szakasz törlődik. Az /n/-nel és /e/-vel felrajzolt alakzatok pontjait nem törli!

/n/ — téglalap rajzolása (NEGY rutin)

Az eljárás megkérdezi az „a" és „b" oldal hosszát. Ez a felosztás egységben értendő. Pozitív „a" és „b" esetén az utolsó pont a négyzet vagy téglalap bal alsó sarka lesz. A program negatív értékekre a megfelelően elforgatott ábrát rajzolja.

/e/ — ellipszis rajzolása (ELLI rutin)

Az eljárás megkérdezi az ellipszis fél nagytengelyének „RX" és fél kistengelyének „RY" hosszát. Az egység az /f/ paranccsal meghatározott felbontás. Ha „RX"="RY", akkor kört rajzol a rutin. Ilyenkor a program az IS-BASIC ELLIPSE utasítását használja. Az utolsó pont az ellipszis közepében lesz, és a rajzolás befejezése után is ide áll vissza.

```
 160 CLEAR SCREEN
 170 SET KEY CLICK OFF
 180 NUMERIC K,L,J,T(1 TO 50,1 TO 2)
 190 FOR I=1 TO 50
 200   LET T(I,1),T(I,2)=0
 210 NEXT
 220 LET F=20
 230 LET K,L,J=0
 240 GRAPHICS
 250 LET N=1:LET KX=640:LET KY=335
 260 SET INK 2
 270 PLOT 1,1;1279,1;1279,719;1279,719;1,
719;1,719;1,1
 280 CALL SZOV
 290 DO
 300   IF KX+K<1 OR KX+K>1279 THEN 360
 310   IF KY+L<1 OR KY+L>719 THEN 360
 320   PLOT KX+K,KY+L;
 330   LET T(N,1)=KX+K:LET T(N,2)=KY+L
 340   LET N=N+1
 350   IF N>50 THEN LET N=1
 360   CALL BOT
 370 LOOP
 380 DEF BOT
 390   GET A#
 400   IF A#="" THEN 390
 410   SOUND PITCH ORD(A#)-93,DURATION 2
 420   SELECT CASE A#
 430   CASE "f"
 440     CALL FELB
 450   CASE "t"
 460     CALL TORL(T,N,KX,KY)
 470   CASE "u"
 480     CALL UJRA
 490   CASE "m"
 500     CALL MOZG
 510   CASE "p"
 520     CALL TOLT
 530   CASE "n"
 540     CALL NEGY(T,N)
 550   CASE "e"
 560     CALL ELLI
 570   CASE "v"
 580     END
 590   CASE ELSE
 600     LET J=JOY(0)
 610     SELECT CASE J
 620     CASE 1
 630       LET K=K+F
 640     CASE 2
 650       LET K=K-F
 660     CASE 4
 670       LET L=L-F
 680     CASE 8
 690       LET L=L+F
 700     CASE 9
 710       LET K=K+F:LET L=L+F
 720     CASE 10
 730       LET K=K-F:LET L=L+F
 740     CASE 6
 750       LET K=K-F:LET L=L-F
 760     CASE 5
 770       LET K=K+F:LET L=L-F
 780     CASE ELSE
 790     END SELECT
 800   END SELECT
 810 END DEF
 820 DEF FELB
 830   CLEAR TEXT
 840   PRINT AT 1,2:"felbontas=";F
 850   INPUT AT 2,11:F
 860   CALL SZOV
 870 END DEF
 880 DEF MOZG
 890   CLEAR TEXT
 900   PRINT "mozgato vege = <blank>"
 910   LET F=F+3
 920   DO
 930     GET S#
 940     SET INK 3:PLOT ;
 950     WAIT .3
 960     SET INK 0:PLOT ;
 970     WAIT .6
 980     SET BEAM OFF
 990     IF KX+K<1 OR KX+K>1279 THEN 1020
1000     IF KY+L<1 OR KY+L>719 THEN 1020
1010     PLOT ,KX+K,KY+L,
1020     IF S#="" THEN 930
1030     IF S#=" " THEN EXIT DO
1040     CALL BOT
1050   LOOP
1060   LET F=F-3
```

```
1070   SET INK 2
1080   CALL SZOV
1090 END DEF
1100 DEF TORL(REF T,REF N,REF KX,REF KY)
1110   CLEAR TEXT
1120   LET M=N-1
1130   PRINT AT 1,3:"<t>=törlés  <blank>=
vege"
1140   SET INK 0
1150   DO
1160     GET A#
1170     IF A#="" THEN 1160
1180     SELECT CASE A#
1190     CASE "t"
1200       IF T(M-1,1)>0 THEN
1210         PLOT T(M,1),T(M,2);T(M-1,1),
T(M-1,2);
1220         LET T(M,1),T(M,2)=0
1230         LET M=M-1
1240       END IF
1250       IF M=1 AND T(50,1)>0 THEN LET
M=50
1260     CASE " "
1270       LET N=M
1280       LET KX=T(N,1)
1290       LET KY=T(N,2)
1300       PLOT KX,KY,
1310       LET K,L=0
1320       EXIT DO
1330     CASE ELSE
1340     END SELECT
1350   LOOP
1360   SET INK 2
1370   CALL SZOV
1380 END DEF
1390 DEF SZOV
1400   CLEAR TEXT
1410   PRINT " f=felbontas t=törles   u=u
j rajz "
1420   PRINT " n=negyszög  e=ellipszis"
1430   PRINT " a=mozgatas  p=festes   v=E
ND"
1440 END DEF
1450 DEF UJRA
1460   FOR I=1 TO 50
1470     LET T(I,1),T(I,2)=0
1480   NEXT
1490   CLEAR GRAPHICS
1500   CALL SZOV
1510   PLOT 1,1;1279,1;1279,719;1279,719;
1,719;1,719;1,1
1520 END DEF
1530 DEF TOLT
1540   CALL MOZG
1550   SET INK 0
1560   PLOT ,FORWARD 3
1570   SET INK 1
1580   PLOT PAINT
1590   SET INK 2
1600 END DEF
1610 DEF NEGY(REF T,REF N)
1620   OPTION ANGLE DEGREES
1630   LET X1=T(N-1,1):LET Y1=T(N-1,2)
1640   PLOT X1,Y1
1650   CLEAR TEXT
1660   PRINT "Kerem a,b-t";
1670   INPUT AX,BY
1680   PLOT 1
1690   FOR I=0 TO 2 STEP 2
1700     PLOT ANGLE I+90;
1710     PLOT FORWARD AX*F;
1720     PLOT ANGLE(I+1)+90;
1730     PLOT FORWARD BY*F;
1740   NEXT
1750   PLOT ;
1760   CALL SZOV
1770 END DEF
1780 DEF ELLI
1790   CLEAR TEXT
1800   PRINT "Kerem rx,ry-t ";
1810   INPUT R1,R2
1820   ! PLOT ,
1830   PLOT ,ELLIPSE R1*F,R2*F
1840   CALL SZOV
1850 END DEF
```

/u/ — új rajz készítése (UJRA rutin)

Az eljárás törli a ponttárolót, letörli a grafikus képernyőt és új keretet rajzol.

/v/ — befejezi a program működését

TOVÁBBI KÉT ELJÁRÁS

A SZOV rutin a grafikus képernyő alatti szöveges lapra írja a menü szövegét.

A BOT rutin a menü parancsait kiválasztó billentyűket és a botkormányt figyeli. Minden billentyű lenyomása és a botkormány mozdítása is más-más hangot ad.

Világszép művünk a program futása után a :VSAVE „képnév" paranccsal háttértárolóra menthető.

FEHÉRVÁRI FERENC

Megkérdeztük az Enterprise-ról

A Centrum Áruházak Vállalat az elmúlt évben közel tízezer darab Enterprise 128 számítógépet adott el. Ezt a típust kizárólag ez a vállalat forgalmazta. Az eddigi tapasztalatokról kérdeztük Kaszás Kálmán vezérigazgatót.

Mikroszámítógép Magazin: Noha a Centrum Áruházak Vállalatnál már hagyománya van a számítógépek értékesítésének, nem kifejezetten erre a területre szakosodtak. Miért döntöttek úgy, hogy nagy mennyiségben, nálunk ismeretlen számítógép forgalmazását kezdik el?

Kaszás Kálmán: Az ilyen tevékenység nálunk nem új keletű. A Commodore 610 típusú géppel kezdtük, ebből hatszázat adtunk el. Amíg a gép ára a hazai piacon hatszázezer forint körül volt, mi jóval kétszázezer alatt kínáltuk. Tehát már akkor is törekedtünk arra, hogy kedvező árat alakítsunk ki. Valós piaci viszonyok között egy kereskedelmi vállalat azt szerzi be, amit forgalmazni kíván. A mi körülményeink között csak abban dönthettünk, hogy a hazai kínálatban megjelent Enterprise számítógépet bevezetjük-e vagy sem. Mi vezérelt minket? Ilyen kategóriájú gépet Magyarországon csak a Videoton gyárt. Ezért vállalnunk kellett a nehézségeket, az ismeretlen márkát, a szervizhálózat kialakítását, a használati utasítással való ellátást, a szoftverek beszerzését. A perifériák lehetőség szerinti jó ellátására is sok energiát fordítottunk. Végül is az üzleti szempont vezérelt minket. Ilyetén a hazai keresletet akartuk megfelelő árú géppel kielégíteni, megteremtve egyúttal a gondos értékesítés körülményeit is. Tízezer Enterprise értékesítésére szerződtünk a Novotrade Rt.-vel, de további háromezret is átvettünk.

M. M.: Ezenkívül nem vesznek át értékesítésre gépeket?

K. K.: Nem. Ennek több oka is van. Az eredetileg szerződött tízezer gép nem a megállapodás szerinti ütemben érkezett. Az engedélyezési eljárás elhúzódása miatt azonban csak 3500-at kaptunk. Futottunk a gép után, hiszen már mindent előkészítettünk. Az ütemtelen szállítás ellenére is átvettünk összesen tizenháromezer Enterprise-t. Úgy véljük, hogy ez a mennyiség a keresletet kielégíti. Importlehetőségünket inkább a kiegészítők és a tartozékok beszerzésére kell fordítanunk. Másik gondunk az, hogy a megváltozott gazdasági szabályozók miatt az új beszerzés nem fér bele a kialakított árba, ami már az átvett utolsó tételre is igaz. Ennek ellenére mi továbbra is tartottuk a számunkra már kedvezőtlen árat, de további tételeket nem veszünk át. A még meglévő mennyiséget a nagykereskedelmi részlegünkön keresztül más kereskedelmi vállalatok számára is értékesíteni kívánjuk. Ebbeli döntésünkben az is közrejátszott, hogy nagyon megnövekedett a számítógépek magánimportja. Bizonyára ez az új útlevélrendeletnek, a többszöri kiutazásnak a következménye.

M. M.: Nem titok, hogy az Enterprise 128 a nyugati piacon megbukott. Igaz, nem a konstrukciója, hanem a rossz marketingtevékenység miatt. Az viszont nyilvánvaló, hogy a vásárlók nem mentesek az előítélettől. Nem vállaltak-e túlzott kockázatot, amikor mindezek ellenére az értékesítés mellett döntöttek?

K. K.: Ismereteink szerint a gép nem szakmailag bukott meg. Ezenkívül azért döntöttünk mellette, mert egy hasonló típus licencét az egyik magyar vállalat megvásárolta. Vállalkozásunkkor számoltunk azzal, hogy egy nálunk ismeretlen márkát kell a vásárlókkal elfogadtatni. A gépet bevizsgáltattuk, és a szakértők nagyon jó véleményt adtak róla. Mivel a tudása és az ára jó volt, ezért szántuk rá magunkat, hogy forgalmazzuk. Így aztán a bevezető reklámmal egyidőben Budapesten kívül még tizenhét helyen kezdtük a gépet eladni. Az eddigi, több mint tízezer értékesítés minket igazol.

M. M.: Megítélés kérdése, hogy az Enterprise 128 szoftverrel, szakkönyvvel és kiegészítőkkel való ellátása jó-e vagy rossz. Az azonban saját tapasztalatunk, hogy nem egyenletes. Miért?

K. K.: Annak az árunak, aminek importtartalma van, annak beszerzése függ az engedélyezési eljárástól. Ez az egész magyar kínálatra jellemző: valamit mindig lehet kapni, de nem folyamatosan. Az áruházak megrendeléseikben önállóak, így a készleteikkel is maguk gazdálkodnak. Úgy érzem, mégsincs sehol sem hiány, mert a nagykereskedelmi raktárunkból folyamatosan ellátjuk az üzleteket. Megemlítem, hogy 1987-ben az eladott tízezer géphez 39 500 játék- és oktatókazetta fogyott el.

M. M.: Hogyan ítéli meg az Enterprise jövőjét? Jogosnak tartja-e azoknak a véleményét, akik a lassú elsorvadását jósolják, illetve mivel akadályozható ez meg?

K. K.: Amint említettem, vállalatunk a már átvett tizenháromezer gépen kívül nem kíván többet beszerezni. Importlehetőségeinket a meglévő gépeknek kiegészítőkkel és tartozékokkal való folyamatos ellátására fordítjuk. Ezt hosszú távon kívánjuk fenntartani, mert a vásárlóinkkal szemben felelősséget érzünk.

M. M.: Tehát az Enterprise-tulajdonosok megnyugodhatnak?

K. K.: Igen. Kérem, ez nálunk nemcsak jogi, hanem erkölcsi kötelezettség is, vállalatunk jó híre függ ettől.

M. M.: Néhány olvasónk azzal vádolja a Centrumot, hogy az Enterprise-t túlzottan preferálja más gépek hátrányára.

K. K.: Először is nagyon meglep ez a „vád". A Videotonnal szerződésünk van, és mi annak továbbra is eleget teszünk. A kínálat bővítésére is változatlanul erőfeszítéseket teszünk. Ezért semmilyen érdekünk nem fűződik a kivételezéshez. Talán az Enterprise reklámozására nagyobb gondot fordítunk — amit az új típus bevezetése indokol —, de még gondolatban sem fordul elő, hogy más gépeket emiatt háttérbe szorítsunk.

PINKE GYÖRGY

MEMÓRIA, GRAFIKA, BILLENTYŰZET

Az alábbi cikket egyik olvasónk küldte be. Levelében azt írja, hogy tapasztalatszerzésében szakirodalomra nem támaszkodhatott, mert külföldön tanul, s ott nem jut hozzá az időközben megjelent könyvekhez. A cikkben öt témával foglalkozik, véleményünk szerint nagyon érthetően, ami a szakkönyvekről sajnos nem mindig mondható el. Reméljük, hogy írásával segítséget nyújtunk az Enterprise-tulajdonosoknak, különösen a kezdőknek.

A MEMÓRIA FELOSZTÁSA ÉS HASZNÁLATA

A memória a gépben 256 darab 16 kbájtos lapra van felosztva. Ezekből alaphelyzetben a ROM és a RAM egyaránt nyolcat foglal el. A maradék 240-et alapkiépítésben a gép nem tartalmazza. A lapokat a SPEEK és a SPOKE utasításokkal érhetjük el. A ROM a Ø—7 lapokat úgy foglalja el, hogy minden lapnak megvan a párja, ahol ugyanaz található. A párok a következők:

Ø — 2
1 — 3
4 — 6
5 — 7

Gondolom, ez a hardver szempontjából azt jelenti, hogy kétszer kapuzza be ugyanazt a lapot a DAVE chip.

A nyolc RAM-lap 248—255-ig található. Bekapcsolás után a 249—253 lapok teljesen üresek, a megmaradó hármat az operációs rendszer használja saját céljaira.

A Z80 mikroprocesszor egyszerre négy lapot tud használni. Az első a 176. portra, a második a 177. portra, a harmadik a 178. portra, a negyedik pedig a 179. portra kapcsolódik. BASIC-ből az OUT n,x utasítással lapozható a memória, gépi kódból pedig a megfelelő OUT utasítással. Az OUT 177,254 utasítás például a 177. portra kapcsolja a 254. lapot. Alaphelyzetben BASIC-ből vizsgálva a lapokat, azok a következőképpen vannak bekapuzva:

Port	Lap
176	248
177	Ø
178	255
179	1

BASIC-ből a 248-as lapot nem ajánlatos lekapcsolni, mert a gép csak RESET-tel indítható újra.

A GRAFIKUS KÉPERNYŐ

A grafikus képernyő helye változik, ha ablakokat nyitunk meg. Bekapcsolás után, német gépen jók az értékek. A grafikus képernyő 14 400 bájtot foglal el a memóriából. Ez a lapokon két részletben helyezkedik el:

1. lap 11308—16383
2. lap Ø—9323

A 254. lap 11308. bájtja a bal felső sarok, a 255. lap 9323. bájtja pedig a jobb alsó sarok. Ha a 177. portra belapozzuk a 254. lapot, akkor a teljes grafikus képernyőre rajzolhatunk POKE utasításokkal. Ekkor a képernyő bal felső sarkába a POKE 27692,x utasítással rajzolhatunk, a jobb alsó sarokba pedig a POKE 42091,x utasítással. Azt, hogy mit rajzolunk és milyen színnel, az *1. táblázat* alapján határozhatjuk meg, a grafikus módtól függően.

A HÍRES GRAPHICS üzemmódban a képernyő egy sora 80 bájtból áll, az ATTRIBUTE módban pedig 40-ből.

GRAPHICS 2. A sor egy bájtja nyolc részre oszlik, és a bájt tartalma kettes számrendszerben ábrázolva jelenik meg a képernyőn. Ahol 1-es áll, ott az INK 1 rajzol, ahol Ø, ott pedig az INK Ø.

Ahhoz, hogy megkapjuk a kirajzolandó pontokat, az 1. táblázat alapján be kell írni az adott bájtba a megfelelő kódok összegét. Például ha a 3. és 5. pozícióra akarunk kitenni egy pontot, akkor 4+16=20 a beírandó szám.

GRAPHICS 4. Ebben az üzemmódban a sor egy bájtja négy kétbites részre oszlik fel. A pontok ekkor is az 1. táblázatban megadott kódok szerint rajzolhatók. Természetesen a táblázat ide vonatkozó részét használjuk. Attól függően, hogy melyik pozíciót milyen színnel akarjuk befesteni, a megfelelő metszéspontokban levő kódszámokat össze kell adni, amit aztán beírunk a kívánt bájtba.

Példa:

1. poz. INK 0 — 0
2. poz. INK 1 — 32
3. poz. INK 3 — 68
4. poz. INK 2 — 8
108

Tehát 108 a bájtba írandó szám.

GRAPHICS 16. Ebben az esetben egy bájt két részre oszlik fel. Ahhoz, hogy a kívánt pontokat a megfelelő színnel befessük, ismét az 1. táblázat ide tartozó része nyújt segítséget. A bájtba írandó számot ugyanúgy határozhatjuk meg, mint a négyszínes üzemmódban.

GRAPHICS 256. Ebben az üzemmódban egyszerűen be kell írnunk a bájtba a kiválasztott szín kódszámát, és ez az egész bájtot olyan színűre festi be. Megjegyzem, hogy azoknak, akik a gépi kódú programozásban jártasak, a kódok bináris alakja sokkal többet árul el, mint a táblázatban használt alakjuk!

GRAPHICS ATTRIBUTE. Attributum üzemmódban a képernyő százhetvenhat 40 bájt szélességű sorból áll. Ez azt jelenti, hogy ahol eddig két bájtot láttunk, ott most egy, kétszer olyan széles bájt található. Ezeket a bájtokat úgy kell elképzelni, mint a GRAPHICS 2 módban: a képernyőn a bájt kettes számrendszerben ábrázolt

1. táblázat

GRAPHICS 2:

Pozíció 8.	7.	6.	5.	4.	3.	2.	1.	Kód
1	0	0	0	0	0	0	0	128
0	1	0	0	0	0	0	0	64
0	0	1	0	0	0	0	0	32
0	0	0	1	0	0	0	0	16
0	0	0	0	1	0	0	0	8
0	0	0	0	0	1	0	0	4
0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0

GRAPHICS 4:

Pozíció 4.	3.	2.	1.	INK Kód
0	0	0	0	0
128	64	32	16	1
8	4	2	1	2
136	68	34	17	3

GRAPHICS 16:

Pozíció 2.	1.	INK Kód
0	0	0
128	64	1
8	4	2
136	68	3
32	16	4
160	80	5
40	20	6
168	84	7
2	1	8
130	65	9
10	5	10
138	69	11
34	17	12
162	81	13
42	21	14
170	85	15

képét láthatjuk, és a bájtok is ugyanúgy programozhatók, mint az előbb említett módban. Ezek a bájtok a memóriából 7200 bájtot foglalnak el (254. lap 11300-tól 255. lap 2123-ig).

A kép kiszínezésére a második 7200 bájt használható. Ezekkel adható meg, hogy milyen legyen a neki megfelelő bájtban a tinta és a papír színe. Színezéskor a palettán megadott színek paletta szerinti számát kell használnunk. A bájt első négy bitje a tinta színét adja meg, a felső négy pedig a papírét.

A színezés bájtjai a 255. lapon 2124—9323-ig találhatók meg. Ha például a képernyő bal felső bájtjába akarunk rajzolni, akkor a 254. lap 11300. bájtjával adhatjuk meg a rajzolatot, a 255. lap 2124. bájtjával pedig a színeket.

2. táblázat

PORT 181:			
x=0		F4	254
Bal SHIFT	127	F5	239
>	253	F6	247
Y	191	F7	223
X	223	F8	253
C	247	x=5	
V	239	8	254
B	251	9	251
N	254	0	239
x=1		8	247
CTRL	127	'	223
LOCK	253	ERASE	191
A	191	x=6	
S	223	J	254
D	247	K	251
F	239	L	239
G	251	Ö	247
H	254	Ä	223
x=2		*	191
TAB	127	x=7	
Q	253	ENTER	191
W	191	ALT	127
E	223	PAUSE	239
R	247	JOY fel	247
T	239	JOY le	253
Z	251	JOY jobb	251
U	254	JOY bal	223
x=3		STOP	254
ESC	127	x=8	
1	253	M	254
2	191	,	251
3	223	.	239
4	247	-	247
5	239	Jobb SHIFT	223
6	251	SPACE	191
7	254	DEL	253
x=4		INS	127
F1	127	x=9	
F2	191	I	254
F3	251	O	251
		P	239

Mi a manó?

A Centrum Áruházakban Enterprise lemezmeghajtó csatlakozó/kontroller kapható. Az eredeti angol csatlakozóból ötszáz darab érkezett, de ha igény lesz rá, importálnak még többet is, illetve megindítják a magyarországi gyártását.

A gépkönyv nem tartalmazza a SET utasítás három opcióját. Ezekkel a lehetőségekkel alkalmazhatjuk mind az ASK, mind a TOGGLE utasítást. Ezek például: TAPE PROTECT a 34. EXOS rendszerváltozót (PROTECT-et) állítja be, amely a kazetta védelmét szabályozza; a PRIORITY a 38. EXOS rendszerváltozót (SPRITE-ot), míg a SOUND STOP a 13. EXOS rendszerváltozót (WAIT_SND) állítja.

A gépkönyvből még két bináris függvény — a BAND és a BOR is — kimaradt, pedig ezek igen hasznosak lehetnek. Amíg például az AND logikai ÉS műveletet végez, vagyis igazán csak az IF utasításban hasznos, addig a BAND, amint a neve is mutatja, bitenkénti ÉS-nek felel meg. Így

5 AND 6=6

5 BAND 6=4

Szintén hiányzik a gépkönyvből az ERNUM és a VERS változó, amely a BASIC-ünk verziószámát, szövegét tartalmazza.

Elterjedt az a nézet, hogy egy sorba nem lehet több BASIC utasítást írni. Ez így nem igaz. Próbáljuk ki:

10 a=0:PRINT A:A=1:PRINT A

Az ún. blokkutasításokat (FOR-NEXT, IF-ENDIF . . .) azonban valóban külön sorba kell írni. A programunk áttekinthetősége érdekében érdemes a többi utasítást is külön sorba írnunk.

A tanácsokat Racskó Tamás és Erdei András olvasóink írták.

A BILLENTYŰZET HASZNÁLATA

A billentyűzetet a processzor a 181. porton keresztül kezeli, tíz csoportra felosztva a gombokat. A csoportok felosztása a *2. táblázatban* található.

Legegyszerűbben az alábbi assembler programmal olvasható a billentyűzet:

```
DI
LD, A,x
OUT (181),A
IN A,(181)
EI
```

Az x a beolvasandó csoport száma, amelyet az A regiszterbe tölt a program, majd lefutás után az A-ban kapjuk vissza az eredményt. Ha nincs billentyű lenyomva, akkor 255-öt kapunk vissza. Ha csak egy billentyűt nyomunk le, akkor azt a táblázat segítségével azonosíthatjuk. Ha egyszerre több billentyű van lenyomva, akkor a megfelelő kódok bitenkénti ÉS kapcsolatának az eredményét kapjuk vissza.

RENDSZERVÁLTOZÓK A MEMÓRIÁBAN

A kézikönyv 162., 193. oldalán közölt rendszerváltozók gépi kódból is könnyen elérhetők. A megadott rendszerváltozók a 255. lapon találhatók a 16325 címtől kezdődően a 16364 címig, ugyanolyan sorrendben, mint a könyvben.

HOL VAN A KARAKTERGENERÁTOR?

A karaktergenerátor 1152 bájtot foglal el, és a 255. lapon található, 13440—14591-ig. A karakterek soronként tárolhatók. Az első 128 bájt a 128 karakter első sorát tartalmazza, a második 128 bájt a második sort és így tovább a 9. sorig. A karakterek programozásánál a kiválasztott bináris kód decimális alakját kell a bájtba beírni.

DÓSA ISTVÁN

Mit tud a QUALIGRAPH?

Szoftverminősítő szoftver

Az előző részt a táblázatok ismertetésével fejeztük be. Folytassuk most az *adatfolyamot jellemző mérőszámkimenetek* ismertetésével. Ilyen szám például a definiálások átlagos értéke vagy a két értékadás közötti programsorok száma. Ha az utóbbi mutatónak kicsi az értéke, az azt jelenti, hogy az azonos változót több célra használjuk. Ez a tapasztalatok szerint nem praktikus, mert könnyen programhibát idézhet elő. Gyakori hiba például, hogy elfelejtünk új értéket adni a változónak. Ha nagy a mutató értéke, akkor arra következtethetünk, hogy fontos változóról van szó, aminek értékét a program több helyen felhasználja.

A kimenetek negyedik csoportja a *kommentanalízis*. A *fejkomment-ellenőrzés* azt vizsgálja, hogy van-e ilyen és megfelel-e a szabványnak. Az adott cég szabványa a QUALIGRAPH indításakor paraméterenként megadható. Ezután megkapjuk a *kommentek gyakoriságát és átlagos távolságát*. Az előbbi az egy programsorra jutó kommentek száma, az utóbbi a két megjegyzés közötti sorok átlagos távolsága. Mind a kettő a forrásprogram áttekinthetőségét, kommentelését jellemzi.

A továbbiakban egy példán mutatjuk be a QUALIGRAPH-ot. Mivel a kimenetek száma még egy igen rövid és egyszerű programnál is meghaladja a közölhető terjedelmet, ezért néhány általunk érdekesebbnek talált táblázatot emelünk ki.

A forrásprogram FORTRAN nyelven készült és úgynevezett logaritmikus rendezést hajt végre *(5. ábra)*. A program két komponensből áll: az LGSORT és a SHIFT elnevezésűből. A két komponens együtt a VALUE, amely egy egydimenziós tömb függvényében rendezi az INDEX nevűt. Ha az INDEX alapértékeiben egytől növekvő számok vannak, akkor megmutatja, hogy a VALUE változók értéke hányadik a sorrendben. A program 49-es sora a rendezés befejezését vizsgálja. Az 50—53-as sor beállítja a segédváltozók kezdőértékeit. A 70-es sor vezérlésátadásából kiderül, hogy a program az *I* változó léptetésével rendez. Az 59-es sor vizsgálja a VALUE értékének sorrendjét.

A SHIFT komponens megcseréli, azaz rendezi az INDEX értékeit. Az algoritmus logikája szerint nem lenne szükség arra, illetve indokolatlan, hogy a SHIFT komponens alprogram legyen. Ez az eljárás szerepelhetne az LGSORT listájában is. A bemutatás kedvéért azonban szándékosan így írták meg a QUALIGRAPH fejlesztői. A forrásprogram további jellemzőit a szoftver kimenetei mutatják meg, hiszen az erre a célra készült. A cikkben szereplő kimenetek az LGSORT komponensre készültek. A táblázatokat mindig a forrásprogrammal egybevetve érdemes nézni, mert a felépítésüket így követhetjük jól.

A hívási gráf rendkívül egyszerű, hiszen csak az LGSORT komponens hívja a SHIFT-et. Az 1. ábra (lásd az előző részt) a bonyolultabb programok vizsgálatát szemlélteti. A hívási és elérhetőségi mátrix *(6. ábra)* is igen kicsi a példánkban. Mégis nagyon fontos információhoz jutunk abból, hogy az LGSORT közvetlenül hívja önmagát, de a SHIFT elkülönült komponens. Példaprogramunkban ezt a 67-es sor végzi el. A bonyolultabb programoknál ez az információ nagymértékben segíti a fejlesztő munkáját. A *7. ábrán* látható kimenet a hívási helyeket mutatja be. A QUALIGRAPH majdnem minden táblázatában a programsor száma mellett a karakterek száma is megtalálható. Ez a megoldás igen hasznos a program javításakor. Ezután sorrendben a hívási szerkezetet jellemző mérőszámok következnek. A minőségmérő számokról az összefoglaló táblázatot közöljük.

```
                  L S
                  G H
                  S I
                  O F
                  R T
                  T

| LGSORT   |- X|
| SHIFT    |  -|
```

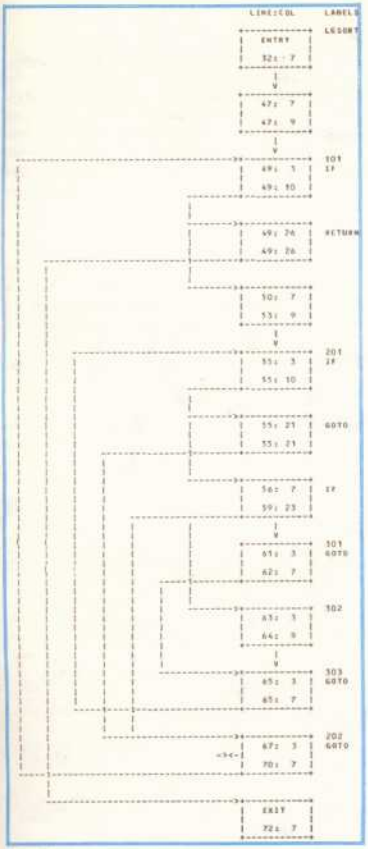
6. ábra

CALLED MODULES	SCOPE	PLACES OF INVOCATION (LINENR : COLUMNR)
SHIFT	EXTERNAL	67 : 7

7. ábra

5. ábra

```
      SUBROUTINE SHIFT ( INDEX , I , J )
C*************************************************************
C*                                                           *
C*    ROUTINE NAME            :   SHIFT                      *
C*    AUTHOR                  :   I. PETROVITS               *
C*    DATE                    :   01/23/1981                 *
C*    FUNCTIONAL DESCRIPTION  :   THE SUBROUTINE MAKES A CYCLICAL*
C*                                SHIFT BETWEEN INDEX(I) AND *
C*                                INDEX(J). INDEX(I)'S NEW PLACE *
C*                                IS THE PLACE DEFINED BY J. *
C*                                                           *
C*************************************************************
C*
      DIMENSION INDEX (1)
      M=INDEX(I)
C
      IF ( I - J ) 21 , 11 , 22
C
   21 K = I+1
      DO  31 L = K,J
   31 INDEX(L-1)=INDEX(L)
      GO TO  11
C
   22 K = I-1
   32 INDEX(K+1)=INDEX(K)
      K=K-1
      IF(K.GT.J) GO TO  32
C
   11 INDEX(J)=M
      RETURN
      END
      SUBROUTINE LGSORT ( VALUE , INDEX , LENGTH )
C*************************************************************
C*                                                           *
C*    ROUTINE NAME            :   LGSORT                     *
C*    AUTHOR                  :   I. PETROVITS               *
C*    DATE                    :   01/23/1981                 *
C*    FUNCTIONAL DESCRIPTION  :   LOGARITHMICAL SORT.        *
C*                                THE ROUTINE SORTS THE INDICLS *
C*                                OF THE VALUES.             *
C*                                                           *
C*    CALLED ROUTINE          :   SHIFT   ( INDEX , I , J )  *
C*                                                           *
C*************************************************************
      DIMENSION VALUE ( LENGTH ) , INDEX ( LENGTH )
      I=2
C
  101 IF ( I.GT.LENGTH ) RETURN
      M=INDEX(I)
      J=I-1
      K=1
      L=J
C
  201 IF ( K.GT.L ) GOTO  202
      J=(K+L)/2
      N=INDEX(J)
C
      IF ( VALUE(M) - VALUE(N) ) 301 , 202 , 302
C
  301 L=J-1
      GOTO  303
  302 J=J+1
      K=J
  303 GOTO  201
C
  202 CALL SHIFT ( INDEX , I , J )
C
      I=I+1
      GOTO  101
C
      END
```



8. ábra

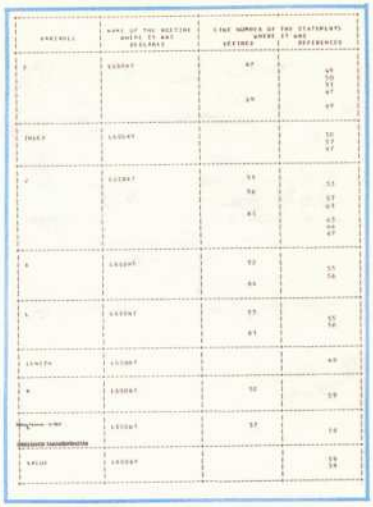
A programfejlesztési és értékelési munkával együtt jár az elágazások áttekintése is. Így a QUALIGRAPH következő kimenete a vezérlési gráf *(8. ábra)*. A forrásprogram alapján láthatjuk, hogy a vezérlésátadás a 70-es sorból a 49-es sorba megy át. Az élek kereszteződései megmutatják, hogy az algoritmus szerinti műveletsor hol ágazik el. A 65-ös sor például három feltételes ugrást „átugrik".A felhasználót segíti, hogy a gráf ábrája a programutasítás címeit is tartalmazza. A vezérlési gráf lényegében megfelel a program folyamatábrájának.

A változók használatának összefoglaló táblázata az egyik „legértékesebb" kimenet *(9. ábra)*. Ebből megtudhatjuk, hogy a változó, illetve változók melyik sorban kapnak értéket, valamint azt is, melyik sorban használjuk fel őket. A bemutató program szerint az *I*-t a 47-es sorban definiáljuk. Ennek értékét a 49, 50, 51, 67 és 69-es sorban használjuk fel. Közben a 69-es sorban az *I* változót megváltoztatjuk, átdefiniáljuk.

A *10. ábra* az adatfolyamot jellemző mérőszámokat tartalmazza. Az első oszlop a változót, a második pedig a komponenst jelöli, mégpedig ott, ahol előfordul. Nézzük meg például az *I* változót. Az LGSORT-on belül a kommenteket nem számolva, a 47-es sor után a 69-es sorban kap értéket. Minimális távolsága nulla, mert egymást követő programsorok használják. A szubrutinhívást figyelembe véve „tartósságát", azaz az értékét maximálisan 22 soron keresztül tartja meg. Ezután az átlag csak osztás kérdése.

Az utolsó táblázatban, amit bemutatunk, a program minőségét jellemző mutatók vannak. A mutatók optimális értékét a QUALIGRAPH felhasználói kézikönyve közli. Az alkalmazók ebből tájékozódhatnak. Akik a dokumentációt nem ismerik, azok a *11. ábrán* láthatnak néhány mutatót.

A hierarchikus komplexitás eggyel egyenlő. Ez természetes, mert egy komponens egyet hív. A demonstrációs program-



9. ábra

ban egy szint van. A strukturális komplexitás értéke 0,5. Két komponens egy hívás. A hívási entrópia nulla, hiszen bizonytalanság nincs. A ciklomatikus szám értéke öt, melyet a vezérlési gráfból láthatunk. Az elérési utak száma szintén öt. Az előírás szerinti kommentek hiányoznak. A fejkommentek száma 19.

A QUALIGRAPH teljes megismeréséhez egy könyv is kevés lenne. A cikkünk célja mindössze az volt, hogy megismertessük olvasóinkat egy, a Magazinban eddig nem szereplő témával, a szoftvert értékelő szoftverrel.

PINKE GYÖRGY

10. ábra

VARIABLES	PLACE OF DECLARATION	DISTANCE MAX	DISTANCE MIN	DISTANCE AVER.	DEF. DURABILITY MAX	DEF. DURABILITY MIN	DEF. DURABILITY AVER.	DEF/REF
I	LGSORT	16	0	3.7	22	22	22.0	0.4
INDEX	LGSORT	16	7	8.5	0	0	0.0	0.0
J	LGSORT	4	0	2.0	7	5	6.0	0.5
K	LGSORT	8	1	4.0	12	12	12.0	1.0
L	LGSORT	5	1	2.7	8	8	8.0	1.0
LENGTH	LGSORT	0	0	0.0	0	0	0.0	0.0
M	LGSORT	9	9	9.0	0	0	0.0	1.0
N	LGSORT	2	2	2.0	0	0	0.0	1.0
VALUE	LGSORT	0	0	0.0	0	0	0.0	0.0
TOTAL	**********	16	0	******	22	0	******	0.5

11. ábra

METRICS	AVERAGE VALUE	MINIMAL VALUE	MINIMAL COMPONENT
HIERARCH. COMPLEXITY	1.00	1.00	LGSORT
STRUCTURAL COMPLEXITY	0.50	0.50	LGSORT
TESTABILITY	0.5000	0.5000	LGSORT
REACHABILITY	1.0	1	LGSORT
AVERAGE REACHABILITY	0.50	0.50	LGSORT
NR OF PATHS/ITERR.NODES	1.00	1.00	LGSORT
AV. LTH OF CALL.PATHS	2.00	2.00	LGSORT
CALLING ENTROPY	0.00	0.00	LGSORT
AV. LTH OF BRANCHES	9.22	9.22	LGSORT
NUMBER OF KNOTS	6.0	6	LGSORT
CYCLOMATIC NUMBER	5.0	5	LGSORT
CONTROL DENSITY	0.3846	0.3846	LGSORT
NUMBER OF STRUCTURES	2.0	2	LGSORT
MAX.NEST OF STRUCTURES	1.0	1	LGSORT
NUMBER OF PATHS	5.0	5	LGSORT
CONTROL ENTROPY	2.24	2.24	LGSORT
AV. LTH OF CTRL.PATHS	10.60	10.60	LGSORT
TOTAL NR OF SRC LINES	40.0	40	LGSORT
NUMBER OF EMPTY LINES	0.0	0	LGSORT
NR OF DECL LINES	1.0	1	LGSORT
NR OF DECL + COMM.	0.0	0	LGSORT
NR OF COMM IN DECL.	0.0	0	LGSORT
NR OF STATEMENT LINES	19.0	19	LGSORT
NR OF STAT. + COMM.	0.0	0	* LGSORT
NR OF COMM IN STAT.	3.42	3.83	LGSORT
MISSING KEYWORDS	100.00	100.00	LGSORT
EMPTY KEYWORDS	0.00	0.00	LGSORT
RAT. OF COMM. IN DCP.	0.00	0.00	LGSORT
RAT. OF COMM. IN STP.	0.27	0.27	LGSORT
TOTAL NR OF OPERATORS	51.0	51	LGSORT
TOTAL NR OF OPERANDS	41.0	41	LGSORT
OBSERVED LENGTH	92.0	92	LGSORT
NR OF UNIQUE OPERATORS	17.0	17	LGSORT
NR OF UNIQUE OPERANDS	11.0	11	LGSORT
VOCABULARY SIZE	28.0	28	LGSORT
CALCULATED LENGTH	107.54	107.54	LGSORT
PROGRAM VOLUME	442.28	442.28	LGSORT
PROGRAM LEVEL	0.0316	0.0316	LGSORT
PROGRAM COMPLEXITY	14012.	14012.	LGSORT
INTELLIGENCE CONTENT	13.96	13.96	LGSORT
MAX.DIST OF VARIABLES	16.0	16	LGSORT
MAX.DEF.DUR. OF VARS	22.0	22	LGSORT
MAX. DEF/REF OF VARS	1.0	1	LGSORT

JÁTSSZUNK KOMOLYAN!

A játék, amint az élet is, sokféle lehet. Ha igaz, hogy élni jó, bátran mondhatjuk: játszani jó. A játék az élet része — szebbik része, talán. És hozzásegíthet ahhoz, hogy életünk más történései is szebbek vagy legalább számunkra értelmezhetőbbek és kezelhetőbbek legyenek. A játék tehát komoly dolog.

Hiszen az élet maga is felfogható egy nagy, folytatólagos játék terepasztalaként, ahol legtöbbünk látszólag szürke figura a kis és nagy Onedin családok küzdelmének tűnő eseménysorban. De tudnunk kell, hogy valamennyien csak annyira vagyunk szürkék és figurák, amennyire ezt a szerepet vállaljuk — hiszen valamennyien önálló, szuverén és valójában fölcserélhetetlen résztvevőként kerültünk a játékba. Kosztolányi *Halotti beszéd*ének szavaival:

> . . . Ilyen az ember. Egyedüli példány.
> Nem élt belőle több és most sem él
> S mint fán se nő egyforma két levél,
> a nagy időn se lesz hozzá hasonló.
>
> . . . a homlokán feltündökölt a jegy,
> hogy milliók közt az egyetlenegy . . .

Önmagunk megtalálását, szuverén személyiségünk, belső függetlenségünk megőrzését segítheti a játék.

Kecskemét a magyar *és* európai jelzővel illethető városaink egyike. Ahhoz, hogy egy város ilyen legyen, sok alkotó gondolat, tettrekészség — tenni akarás és tenni tudás —, kitartó egyéni és közös munka szükséges. Talán az sem véletlen, hogy épp itt létesült a város sajátos képét tovább gazdagító *Játékmúzeum*. Ez a hazánkban egyedülálló intézmény — pontos nevén: Szórakaténusz Játékműhely és Múzeum — 1981 decembere óta működik. Célja az, hogy egyrészt *gyűjtse* és *bemutassa* az ország, az egész magyar nyelvterület, valamint más népek játékait, másrészt *megismertesse* és *megszerettesse* a fiatalokkal és pedagógusokkal a régebbi és az újabb játékokat, továbbá az ősi kézműves hagyományokat.

A vásárlás és főleg ajándékozás révén egyre gyarapodó múzeumi gyűjteményt mintegy tízezer tárgy alkotja. Itt vannak a hazai játékgyártás történeti múltját jelentő műhelyek eredeti termékkollekciói, napjaink hazai gyártmányú játékai, a kiemelkedő gyermekszakkörök alkotásai, országos (gyermekrajz-, játék- és babakészítő-) pályázatok díjnyertes anyagai. Természetesen az itthon elterjedt, külföldi gyártmányú játékok is. A más népek játékait tartalmazó gyűjteményt főleg az ide látogató külföldi szakemberek ajándékai gazdagítják.

Az épületbe lépőt a hazai játéktörténet néhány nagy egyéniségének portréja fogadja: *Kempelen Farkas* (1734—1804), akire azért illik úgy emlékeznünk, hogy *mellesleg* csinált egy világhírű sakkozógépet is, hiszen rendkívül sokoldalú tudós és szervező volt: ő vezette a budai királyi palota építkezéseit, irányította a nagyszombati egyetem Budára telepítését, új, erőátvitelre alkalmasabb, körmozgást végző gőzgépkonstrukcióit készített, ő tervezte a schönbrunni palota kertjének szökőkútjait és a pozsonyi hajóhidakat, míg beszélőgépével és az emberi beszéd mechanizmusát ismertető könyvével a fiziológiai hangtan egyik megalapozója lett; *Brunszvik Teréz* (1775—1861), az első magyar kisdedóvó létrehozója; *Kiss Áron* (1845—1908), pedagógus, a magyar népi gyermekjátékok gyűjtésének első szervezője; *Neumann János* (1903—1957), matematikus, aki nagyszabású életműve részeként a számítástechnikában és a játékelméletben is maradandót alkotott, valamint *Rubik Ernő* (sz. 1944), építészmérnök, a „bűvös kocka" és több más logikai játék kitalálója.

A jelenlegi állandó kiállítás három részből áll. Az elsőben a játszó személytől aktív tevékenységet kívánó, ún. konstrukciós *építőjátékok* vannak. Ezek alapelemei, anyaguk és kialakításuk szerint is igen sokfélék lehetnek. A fa építőkockáktól a fém (például Märklin) és a műanyag (például LEGO) építőszekrényekig számos megoldással találkozhatunk. Szintén nagy csoportot alkotnak a papírjátékok, amelyek régebben az iskolai oktatásban is komoly szerepet kaptak. A szín- és formaérzék fejlesztését szolgálták az üveg-, gyöngy-, színezett kő- és kerámiakirakók.

A kiállítás másik két része a városi, illetve a falusi gyerekek tárgyaival foglalkozik „Nagyszüleink játékai városban és falun" címmel. (Természetesen ez a besorolás a rendszerezést megkönnyítő, lazán értelmezett kategória.) Mind a szabadtéri, mind a szobai időtöltést tekintve gyakran csak a felhasznált anyagban és a készítés módjában van különbség a városi és a falusi gyerekek játékai között. Így például gyári gumilabda és házi készítésű rongylabda; porcelán alvóbaba, illetve a helyi népviseletet tükröző, szőttesmaradékokból, csuhéból csinált baba; sorozatban gyártott textil- vagy felfújható gumiállatok és fából, agyagból otthon eszkábált játékállatkák. Láthatunk itt babaházakat — konyhákat, szobákat, kertet —, közlekedési eszközöket, a fejlődés sorrendjében kis szekeret, vonatot, autót, ólomkatonákat, játékkardot, puskát, foglalkozásokat, szakmákat megkedveltető, készségfejlesztő kellékeket (nyomdák és társaik), valamint különféle társasjátékokat: táblás játékokat és kártyákat. (A kézirat leadása óta új állandó kiállítás is nyílt „Játék és ünnep" címmel. *A szerk.*)

De a látványon túl az itteni műhelymunka adja meg az intézmény igazi jelentőségét. Megismertet a játékok történetével, készítésével — a játékmesterek „belevisznek" a játékba, és miközben játszol, miközben formálódik ujjaid nyomán az anyag, megízleled az alkotás örömét. Úgy találkozhatsz itt a természettel, hogy felfedezed a bőr, a fa, az agyag rejtett szépségeit, amelyeket te tudsz — csak te! — kihozni belőle, ha akarod. És közben talán megsejthetsz valamit abból is, hogy a szépség titka benned is ott van — kár lenne, ha veszni hagynád, sok rárakódott szellemi salak alatt . . .

És miközben magadra találsz, megtalálhatod a társaidat is. A hagyományok, az ünnepekhez kötődő népszokások felelevenítése, megújítása ráébreszthet arra, micsoda roppant érték — és erő! — van egy nép múltjában: ebbe kapaszkodhatsz, ha formálni akarod a jövőt!

●

. . . Újsághír: videózott, iszogatott, majd mellbe lőtte az apukát; lelőtte és kirabolta a benzinkutast . . . — micsoda sivárság, bennük és körülöttük. Lehetséges itt még játszani? Kell!!

●

1987 májusában megalakult a *Kiss Áron Magyar Játék Társaság*, amely önálló jogi személyként, kecskeméti székhellyel, de az egész országra kiterjedően végzi munkáját. Szorosan együttműködik a szakterületet művelő intézményekkel, elsősorban a Szórakaténusz Játékmúzeummal, amely a tevékenység szervezeti, intézményi hátterét képezi.

A társaság a hazai játék- és játszáskultúra értékeinek feltárására, megőrzésére és gazdagítására, az e tárgykörrel foglalkozó szakemberek és érdeklődők összefogására, a játékkultúra sokirányú és erőteljes társadalmi hatásának (és ebből adódó fontosságának) elismertetésére vállalkozott. A társaság — tudva azt, hogy a játék társadalomformáló tényező — kötelességének érzi, hogy sajátos eszközeivel elősegítse a humán jellemvonások, az alkotókészség és a nemzeti önismeret fejlesztését szolgáló, valamint a családközpontú nevelést segítő, a szülő-gyermek kapcsolatot erősítő játékok kidolgozását, bemutatását és elterjesztését.

A tevékenység négy szakosztály keretében folyik; ezek a következők.

Játékszerek és történetük. Ide tartozik az eszközök gyűjtése, rendszerezése, korok és kultúrák szerinti feldolgozása; a különböző népek és társadalmi csoportok játékszokásainak, hagyományainak feltárása; a témához kapcsolódó technológia- és ipartörténet.

Játékok a nevelésben. A játéktevékenység vizsgálata gyermekek és felnőttek körében. Játékok felhasználása a gyermekek pozitív tulajdonságainak alakításában, személyiségük formálásában. Játékok alkalmazása a honismeret elmélyítésére, nemzeti értékeink, valamint más népek és értékeik bemutatására. Játékok gyógypedagógiai és terápiai alkalmazása. Az iskolai tananyag elsajátításának megkönnyítése, nyelvi, művészeti stb. készségek fejlesztése játékokkal. Az öntevékeny játék és a játékkészítés szerepe a személyiség fejlődésében.

Korunk játékai. A korunkra jellemző játékok, elsősorban a műveltségi, sport-, elektronikus/számítógépes és a különféle konstrukciós játékok megismerése és feldolgozása. A gyerekek és a felnőttek játékai. A játszási szokások és társadalmi hatásaik, a szabadidő játékos eltöltésének helyzete. A tömegkommunikáció, a könyv- és folyóiratkiadás szerepe a játékok bemutatásában és népszerűsítésében. A játékok várható fejlődése.

Játékfejlesztés, -gyártás és -terjesztés. Játékszerek kitalálása és védelme (szabadalmaztatás, mintaoltalom), előállítása (tervezés, gyártás, csomagolás; anyag- és gyártástechnológia; szabványosítás, minőség-ellenőrzés), valamint kereskedelmi, forgalmazási (hazai terjesztés, export, import) és egyéb szolgáltatási kérdések (például javítás, garancia). Amatőr játékkészítés.

Itt ez az alighogy megszületett társaság, és amint láthatjuk, tiszteletre méltó a program. Nyitott szervezet — bármelyikünk hozzájárulhat ahhoz, hogy az elgondolásokból minél előbb, minél több értékes eredmény szülessen.

Furcsa dolog ez: higgadtan játékról beszélgetni, miközben a földhözragadt szegénység és a hivalkodó jólét kettős szorítását, durva versenykihívását érezzük, percenként tapasztaljuk. A családok anyagi helyzetének különbsége és a lehetőségek ebből fakadó eltérése mind a felnőttek, mind a gyerekek számára állandó nyugtalanság forrása. Az eszközök birtoklása indokolatlan fölényérzethez, hiányuk kisebbségi érzés kialakulásához vezethet. Az eltorzult értékarányok helyreállítása és a feszültségek csökkentése érdekében a technika elvárázsolt kastélyaiból ki kell csalogatnunk, a „szellemi csócsa" zuhatagából ki kell menekítenünk és előtérbe kell állítanunk *az embert*. Tudatosítanunk kell, hogy például a számítógép és a video nagyszerű segédeszköz lehet, de veszélyes és kártékony is, ha torz felfogásban használják. Tudatosítanunk kell, hogy az a gyerek, aki az apjától megtanulta, hogyan csinálhat gesztenyéből és faágakból bábukat, az együttlét és a természethez fűződő kapcsolat erősítése révén nagyobb és maradandóbb kincset kapott, mint az, akinek odadobtak néhány legújabb kazettát. Az öncélú szerzésnek ára van — gondoltál-e arra, mit veszítesz azzal, ha közben saját magad veszíted, vesztegeted el, ha a „kaparás" közben elszakadsz a környezetedtől, volt és lehetséges barátaidtól?

●

Széthullott családok, elmagányosodott gyerekek és felnőttek között járunk: netalán, sőt nagy valószínűséggel mi is közéjük tartozunk. Meg kell kérdeznünk magunktól: mit tekintünk értéknek, miért mit adtunk cserébe? Milyen „vetélkedőbe" ugrottunk bele vagy hagytuk beugratni magunkat? Saját harmonikus egyéniségünk, belső békénk megőrzése vagy visszaszerzése — ez az életünk igazi nagy játéka. Mert önmagunk, hagyományaink feladása, a megalkuvás oda vezet, ahol most vagyunk. A belső szabadság a külső kibontakozás kulcsa. Csak a gyökértelen ember manipulálható. De nem vagy gyökértelen, ha úgy akarod: szűkebb környezeted (tőled is függően szorosabb vagy lazább) természetes kötődései mellett beletartozol a magyarul beszélők határokkal nem korlátozható közösségébe és az emberiség egyenjogú népekből álló nagy családjába. Itthon vagy ezen a Földön — ami a „családtag" kétes biztonságát jelenti és kötelességeit rója rád. A pénz és a hatalom Onedinjeivel szemben adjon erőt az a tudat, hogy *nem mindennapi társasjáték* részese lehetsz: népünket kell — sokadszorra — lelkileg és fizikailag is feltámasztanunk, közben azon is munkálkodva, hogy ne az egymásra uszított emberek gyűlöletének lángja feszítse szét, hanem a béke melege forrassza össze rozzant bolygónkat.

Ismerjük meg, tanuljuk meg és ne keverjük össze a játékszabályokat!

SIPKA LÁSZLÓ

DIGITAL szelvény
Mikroszámítógép Magazin
1988. augusztus

Egy sarokkal olcsóbb!!

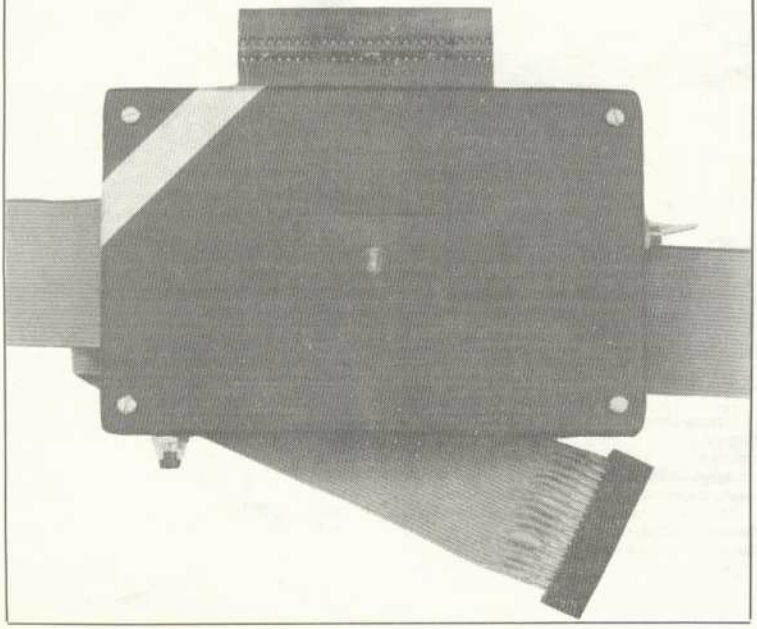
A DIGITAL Számítástechnikai Szaküzlet (Budapest, Szilágyi Erzsébet fasor 35. 1026) Sinclair-termékekre szakosodott: elsősorban a ZX81-es és Spectrum gépekhez használatos eszközöket, programokat árul. De kaphatók itt más gépekhez való tartozékok (például botkormányok), számítástechnikai alkatrészek (integrált áramkörök stb.) és zsebszámológépek is.

Aki ebben az üzletben a lapunkból kivágott sarokszelvényt átadja, vagy megrendelésével együtt oda elküldi, minden hónapban más-más cikket olcsóbban vásárolhat meg. A kedvezmény a szelvényen feltüntetett hónapban érvényes. Minden árengedménnyel vásárolt darabhoz le kell adni egy szelvényt.
A bolt utánvéttel szállítást is vállal, és a szokásos 6 hónap helyett *1 év garanciát ad.*

A DIGITAL
Számítástechnikai Szaküzlet e havi, s egyben utolsó kedvezménye:

Floppy-interfész.
Ára 14 000,— Ft, engedmény 8,5%.

Az illesztő alkalmas Spectrum számítógép és SA400 kompatibilis lemezegységek összekapcsolására. Ezen az egyoldalas, egyszeres sűrűségű lemezegységen kívül a Shugart cég kétoldalas hajtóit is kezeli. A választékhoz mind az egy–, mind a kétoldalasnál kétszeres sűrűségű egység is tartozik. Az illesztő minden fajtából több (összesen legfeljebb négy) hajtót is képes működtetni. Így kétoldalas, kétszeres sűrűségű egységeket használva a kezelt maximális háttértár 2,6 Mbájt.



ISKOLASZÁMÍTÓGÉP-SZERVIZ

Budapest VII.,
Baross tér 19. 1077
Telefon: 428-999

Vállalja:
IBM PC/AT, IBM PC/XT és Commodore típusú (C16, C Plus/4, C64, C128) gépek javítását, átalánydíjas szervizét,

egyedi programok, programcsomagok készítését.

Minden kedden 17-től 20 óráig
HCC ENTERPRISE klub
a VSZM
Közösségi Házban
(Bp. XI., Fehérvári út 120.)
Klubvezető: Romvári Gábor
Telefon: 810-950/473

Közprogramok

Olvasóink egyre gyakrabban küldenek be olyan hasznos programokat, amelyeket terjedelmi okok miatt nem áll módunkban közölni. Mivel azonban úgy véljük, hogy közérdeklődésre tarthatnak számot, ezeket röviden ismertetjük. Akik használni szeretnék a programokat, a részletes leírást, programlistákat vagy — ahol erre mód van — a programot adathordozón, levélben megrendelhetik szerkesztőségünktől. A listák, ill. adathordozók másolását a KASZKAD Kisszövetkezet óbudai Pólus Szakcsoportja végzi (lásd lapunk 1988/2. számát!). A másolási díj listák, leírások esetén oldalanként 8,— Ft, kazetta esetén 150,— Ft, floppy esetén 300,— Ft. Az adathordozóra történő másolás díja az adathordozót is magában foglalja.

Közületektől cégszerű megrendelést kérünk, a másolási díjat az MNB 208-42518-7014 számlára kell befizetni. Magánszemélyek a díjat a KKVMF Pólus Szakcsoport, Bp., Bécsi út 94—96. 1034 címre fizethetik be.

A megrendelés az alábbi formában lehetséges:

MIKROSZÁMÍTÓGÉP MAGAZIN SZERKESZTŐSÉGE
1371 Budapest, Pf. 433.

Megrendelem a Mikroszámítógép Magazin 1988/... számában szereplő közprogramok közül az alábbiakat:
A program neve
Csak programlistát és leírást kérek
Példányszám
Csak adathordozót kérek
Mindkettőt kérem

A megrendeléshez csatolom a szolgáltatási díj befizetését igazoló csekkszelvényt. Dátum, név, pontos cím.

A programok ellenőrzése nem áll módunkban, ezért az esetleges programhibákért mi nem vállalhatjuk a felelősséget.

Kérjük, akik saját készítésű programjaikat — díjtalanul — felajánlják a köz javára, a részletes programleírást, a listát, az adathordozót és az alábbiakhoz hasonló röviditett ismertetőt juttassák el címünkre. (A megrendelők számára nevük jelenti a garanciát.)

664 blokkra rendező program

Programnév:	nincs külön név
Géptípus:	C64
Konfiguráció:	gép és lemezegység
Adathordozó:	lemez
Terjedelem:	132 BASIC-sor
A készítő neve:	Lucz Géza
Megjegyzés:	nincs
A leírás oldalszáma:	1
A program oldalszáma:	3

A program a lemezek jó helykihasználását segíti. Lemezt rendezhetünk a nem kívánt programok törlésével, a kívántak névváltoztatásával, új nevek és blokkszámok beírásával.

A többrészes programok egy gyűjtőnévvel láthatók el.

A rendezett programok képernyőre vagy nyomtatóra listázhatók.

Egy egyszerű nyilvántartó

Programnév:	VIDEOKATALÓGUS
Géptípus:	C16, Plus/4
Konfiguráció:	kazettás tároló és gép
Adathordozó:	kazetta
Terjedelem:	76 BASIC-sor
A készítő neve:	Halmai Csaba
Megjegyzés:	a gép bővítő használatát nem igényli
A leírás oldalszáma:	1
A program oldalszáma:	2

Néhány száz nyilvántartandó valami (példaként videofelvételek) kikeresésére, listázására, az egyes tételek átírására szolgál.

Turbotöltő használata nélkül csak kevés adat esetén célszerű használni.

Mandelbrot-halmazt előállító program

Programnév:	Mandi
Géptípus:	IBM PC/AT és kompatibilis gépek
Konfiguráció:	nincs megadva
Adathordozó:	nincs megadva
Terjedelem:	356 sor Turbo Pascal
A készítő neve:	Nagy Sándor
Megjegyzés:	Spectrum-változat is van
A leírás oldalszáma:	3
A program oldalszáma:	4

A lapunkban eddig megjelentektől — a géptípuson kívül — a képelőállítás gyorsabb módjában tér el. A gyorsítás azon alapszik, hogy a halmazt lerajzolva elvileg egyetlen foltot kapunk; ezen belül a legnagyobb a műveletigénye egy pont kiszámításának, mivel a számítás itt mindig eljut az integráció határáig. Ez a folt körüljárható, és berajzolva a határvonalat, befesthető, amivel elkerülhetjük az így befestett pontok kiszámítását.

Máriáss László rajza

Sorozatunkban azokat az új hardver- és szoftvertermékeket ismertetjük, amelyek várhatóan általánosan elterjednek, és meghatározó szerepük lesz a fejlődés irányainak kialakításában.

Új és hagyományos háttértárolók

A kompaktlemez-lejátszók gyártói igyekeznek „minden piacon eladható termékkel" kirukkolni. Ezért ezek az eszközök nemcsak nagy kapacitású adattárolók, hanem lemezjátszók is.

Sony CD–ROM hajtó

Egy korábbi cikkünkben (1987/2. szám, 39. oldal) írtunk arról, hogy a Sony és a Philips cégek szabványositani kivánják a CDI-ROM (interaktiv kompatlemez-tároló) termékeket. A cég CDU-7101 típusú önálló egységet alkotó CD-ROM hajtója nem tartozik ezek közé, többek között azért sem, mert nem a Microware cég OS-9 operációs rendszerét használja, hanem egy MS-DOS kiterjesztést (CD-ROM Extension). A rendszer része egy PC-illesztő — az ún. Sony bus — lemezkezelő, a hajtó és természetesen a kábelezés. A tárolt adatok mennyisége 540 M lemezenként, ami egyaránt lehet adat vagy zene.

Az adatátviteli sebesség 150 kbit/s, az egyenkénti átvitelnél és ha az átvitel blokkonkénti (2048 vagy 2936 bájt), akkor 600. Az átlagos hozzáférési idő 0,5 s. Az egységet többfajta, a lemezek mechanikai sérülését akadályozó védelemmel látták el. A lemezt kézzel, számítógépvezérléssel és vészhelyzetben automatikusan lehet kivenni. Beépitett hibavédelme egyrészt tehermentesíti a számítógépet, másrészt megnöveli a hatékony adatátviteli sebességet. A Sony-buszon keresztül négy egység láncolható össze.

Atari kompaktlemez-lejátszó

Az előzőhöz hasonló kapacitású tároló átviteli sebessége ST, vagy (az új) Mega számítógépet használva elérheti a 10 Mbit/s értéket is. Az egységhez közvetlenül csatlakoztatható mikrofon és fejhallgató, ha hagyományos célra akarjuk használni. Ezért az egységet egy, a lemezjátszó- és tévétechnikában elterjedt távvezérlővel is ellátták, de vezérelhető a számítógépről is.

Maxtor optikai lemezegység

Optikai tárolókból az elmúlt évben hozzávetőlegesen egymilliárd dollár értékűt adtak el. Ez az összeg a becslések szerint 1990-ben már kilencmilliárd lesz. Eddig a WORM (write once, read multiple — egyszer irható, többször olvasható) lemezek terjedtek el, de ebben az évben várható, hogy a törölhető lemezek tömeggyártását is elkezdik. Ezek az eszközök mikrofilmek, szalagos tárolók helyettesítésére terjedtek el. A cserélhető lemezes változatok a winchester-tárolókhoz nagy kapacitású back-up egységként is használhatók, de jók például röntgenfelvételek tárolására is. Jobbak a hagyományos archivumoknál, illetéktelenek nehezebben férhetnek hozzá és kisebb a helyigényük is.

A Maxtor cég a beirást egy kis teljesítményű lézerdiódával, hőhatással végzi. A felirást CLV (constant linear velocity — állandó lineáris sebesség) technológiával készítik. Az RTX-800S hajtó és az OC-800 kazetta *(1. kép)* egyaránt 800 M tárolókapacitású, kétoldalas. Jellemzői az *1. táblázatban* láthatók.

1. kép

2. kép

3. kép

A Maxtor cég

1982-ig nagy kapacitású lemezegységnek a 30 M feletti tárolókat tekintették. Ekkor jelent meg ez a cég egy 140 M kapacitású, 5" winchester-tárolóval. A „nagy ugrást" olyan technológiai megoldásokkal érték el, amelyekkel egykártyás vezérlést, kisebb iró/olvasó fejet, jobb motorelhelyezést alkalmazhattak. A cég forgalma egy év alatt háromról százötven millióra (!!) nőtt. Eleinte a tárolókapacitást 380 M-ig növelték, mert ezt tartották elsődlegesnek, majd a keresési időt csökkentették. A cég az illesztők szabványositásának is előharcosa. Így vált javaslatára szabványná az ESDI (Enhanced Small Device Inter-

Adatok	RXT800S	OC800
Formált kapacitás		
M	800	800
Átviteli sebesség		
Mbit/s	1,25	
Hozzáférési idő ms		
Átlagos	108	
Sávtól sávig	4	
Maximum	324	
Fordulatszám	11,13	
Késleltetés	60	
Feliratozási		
sűrűség bpi	30200	
Sávsűrűség tpi	15900	
Szektorszám	191952	
Megjegyzés	Spirál-szektorok	
Tárolási idő év		10

1. táblázat

Winchester-illesztők és Maxtor hajtók

A számítógép-háttértároló kapcsolat létrehozásához kétféle illesztőcsoport terjedt el. Az első a hajtó szintű, amelynél két illesztő van: egy a hajtóban (például ST506/412 vagy az ESDI), és ehhez kell készíteni egy illesztőt a számítógépben. Ezeknél a gyártóknak kell elkészíteni az egyik, a felhasználónak a másik illesztőt. Ezeknek az illesztőknek általában szorosan kötöttek a jellemzőik, ezért kevéssé illeszkedhetnek a felhasználók speciális igényeihez.

A másik csoport a számítógép szintű (például SCSI). Itt egyetlen illesztő van, a számítógépben. Ezek intelligens, széles variációs lehetőségű illesztők. Egyes esetekben, ha a gép eredetileg nem készült ennek elhelyezésére, külső illesztőt csatlakoztatnak a számítógéphez.

4. kép

Adatok	XT1140	XT2190
Kapacitás, formálatlan		
M	143,42	191,23
M felületenként	9,56	12,74
sáv	10416	10416
Kapacitás, tipikus formált (512 bájt-szektor)		
M	119,85	159,8
M felületenként	7,99	10,65
sáv	8704	8704
szektor	512	512
szektor/sáv	17	17
Átviteli seb. Mbit/s	5	5
Keresési és léptetési idő ms		
sávtól sávig	4	4
Maximum	46	52
Fordulatszám	60	60
Hozzáférés ms	8,33	8,33
Feliratozási		
sűrűség bpi	9280	11155
Sávsűrűség tpi	1070	1070
Hengerszám	918	1224
Sávszám	13770	18360
Szektor (0,5k)	234090	312120
Adatfej	15	15
Szervofej	1	1
Lemez	8	8

2. táblázat

Adatok	XT1120R	XT1240R
Kapacitás, formálatlan		
M	127,99	239,98
M felületenként	15,99	15,99
sáv	15624	15624
Kapacitás, tipikus formált (512 bájt-szektor)		
M	104,85	196,6
M felületenként	13,1	13,1
sáv	12800	12800
szektor	512	512
szektor/sáv	25	25
Átviteli seb. Mbit/s	7,5	7,5
Keresési és léptetési idő ms		
sávtól sávig	4	4
Maximum	49	49
Fordulatszám	60	60
Hozzáférés ms	8,33	8,33
Feliratozási		
sűrűség bpi	14901	14901
Sávsűrűség tpi	1070	1070
Hengerszám	1024	1024
Sávszám	8192	15360
Szektor (0,5k)	204800	384 000
Adatfej	8	15
Szervofej	1	1
Lemez	5	8

3. táblázat

face — kiterjesztett, kis készüléket illesztő), és ők alkalmazták elsőként az SCSI (Small Computer System Interface — kisszámítógépes rendszer illesztője) szabványt is. Ezekről a szabványokról később lesz szó.

Az optikai tárolók gyártását szintén az elsők között kezdték el. Olyan szigorú statisztikus minőségellenőrzési rendszert alkalmaznak a gyártásban, hogy a legnagyobb vevőik elhagyták a beérkezéskor esedékes minőségellenőrzést, ami ott — a nálunk szokásostól eltérően — gyakorlatilag kötelező.

A cég első termékei, az XT1000 és 2000 sorozat egy SCSI „híddal" csatlakoztathatók az IBM PC Bus vagy Multibus valamelyikéhez. Adataikat a *2.* és *3. táblázat* tartalmazza.

A későbbi ESDI illesztős termékekhez is kell további illesztő. Ezek az XT4000E és 8000E, amelyek adatait a *4.* és *5. táblázatban* foglaltuk össze.

A legújabbak, az XT4000S és 8000S (*6.* és *7. táblázat*) már az SCSI szabványhoz készültek. Ezek nagy kapacitású és igen gyors egységek.

A Seagate cég

A cég terméke az ST225, az itthon és világszerte legismertebb winchester a PC-khez, de a cégről ennél többet kevés olvasó tud.

A cégalapító Alan F. Shugartot az Electronics Business című lap, 2500 olvasójának véleménye alapján, kategóriája legjobb cégirányítójának tartotta 1987-ben (olyanok társaságában, mint például a magyar származású Andrew S. Grove, az Intel elnöke). Ki ő?

Az IBM-nél kezdte, és itt tizennyolc év alatt a rendszerfejlesztési részleg műszaki igazgatói rangjáig vitte. Innen a lemezeiről, szalagjairól ismert Memorex céghez ment alelnöknek, ahonnan állítólag azért vált meg, mert ez a cég nem látott fantáziát az általa szorgalmazott hajlékonylemezes egységek gyártásában. Shugart ezek után megalapította a Shugart céget. Itt

Adatok	XT4170E	XT4380E
Kapacitás, formálatlan		
M	179,41	384,46
M felületenként	25,64	25,64
sáv	20940	20940
Kapacitás, tipikus formált (512 bájt-szektor)		
M	157,93	338,41
M felületenként	22,56	22,56
sáv	18432	18432
szektor	512	512
szektor/sáv	35	35
Átviteli seb. Mbit/s	10	10
átlagos	16	18
Keresési és léptetési idő ms sávtól sávig	3	3
Maximum	36	36
Fordulatszám	60	60
Hozzáférés ms	8,33	8,33
Feliratozási sűrűség bpi	21064	21064
Sávsűrűség tpi	1070	1070
Hengerszám	1224	1224
Sávszám	8568	18360
Szektor (0,5k)	308448	660960
Adatfej	7	15
Szervofej	1	1
Lemez	4	8

4. táblázat

5. táblázat

Adatok	XT830E	XT8760
Kapacitás formálatlan		
M	410,09	768,92
M felületenként	51,26	51,26
Sáv	31410	31410
Kapacitás, tipikus formált (512 bájt-szektor)		
M	360,97	676,82
M felületenként	45,12	45,12
sáv	27648	27648
szektor	512	512
szektor/sáv	54	54
Átviteli seb. Mbit/s	15	15
Keresési és léptetési idő ms sávtól sávig	3	3
Maximum	35	35
Fordulatszám	60	60
Hozzáférés ms	8,33	8,33
Feliratozási sűrűség bpi	31596	31596
Sávsűrűség tpi	1376	1376
Hengerszám	1632	1632
Sávszám	13056	24480
Szektor (0,5k)	705024	1321920
Adatfej	8	15
Szervofej	1	1
Lemez	5	8

Adatok	XT4280S	XT4380S
Kapacitás, formálatlan		
M	281,94	384,46
M felületenként	25,64	25,64
sáv	20940	20940
Kapacitás, tipikus formált (512 bájt-szektor)		
M	248,17	338,41
M felületenként	22,56	22,56
sáv	18432	18432
szektor	512	512
szektor/sáv	36	36
Átviteli seb. Mbit/s	4	4
Keresési és léptetési idő ms sávtól sávig	18	18
Maximum	3	3
Fordulatszám	36	36
Hozzáférés ms	60	60
Feliratozási sűrűség bpi	8,33	8,33
Sávsűrűség tpi	21064	21064
Hengerszám	1070	1070
Sávszám	1224	1224
Szektor (0,5k)	13464	18360
Adatfej	11	15
Szervofej	1	1
Lemez	6	8

6. táblázat

7. táblázat

Adatok	XT8380S	XT8760S
Kapacitás, formálatlan		
M	410,9	768,92
M felületenként	51,26	51,26
sáv	31410	31410
Kapacitás, tipikus formált (512 bájt-szektor)		
M	360,97	676,82
M felületenként	45,12	45,12
sáv	27648	27648
szektor	512	512
szektor/sáv	54	54
Átviteli seb. Mbit/s	4	4
Keresési és léptetési idő ms sávtól sávig	3	3
Maximum	35	35
Fordulatszám	60	60
Hozzáférés ms	8,33	8,33
Feliratozási sűrűség bpi	31596	31596
Sávsűrűség tpi	1376	1376
Hengerszám	1632	1632
Sávszám	13056	13056
Adatfej	8	15
Szervofej	1	1
Lemez	5	8

8. táblázat

Adatok	ST125	ST138
Kapacitás, formálatlan		
Mbájt	25,6	38,4
M felületenként	41,664	62,496
sáv	10416	10416
Kapacitás, tipikus formált (512 bájt-szektor)		
Mbájt	21,4	32,1
K felületenként	2460	3690
sáv	8704	8704
szektor	68	102
szektor/sáv	17	17
Átviteli seb. Mbit/s	5	5
Keresési és léptetési idő ms sávtól sávig	8	8
Maximum	70	70
Fordulatszám	60	60
Hozzáférés ms	8,3	8,3
Feliratozási sűrűség bpi	15500	15500
Sávsűrűség tpi	824	824
Szektor (0,5k)	615	615
Szervofej	4	6
Lemez	2	3

Adatok	ST138R	ST157R
Kapacitás, formálatlan		
M	38,4	57,7
M felületenként	62,496	93,744
sáv	15624	15624
Kapacitás, tipikus formált (512 bájt-szektor)		
M	32,7	49,1
K felületeként	2460	3690
sáv	13312	13312
szektor	104	156
szektor/sáv	26	26
Átviteli seb. Mbit/s		
Átlagos	7,5	7,5
Keresési és léptetési idő ms sávtól sávig	8	8
Maximum	70	70
Fordulatszám	60	60
Hozzáférés ms	8,3	8,3
Feliratozási sűrűség bpi	23250	23250
Sávsűrűség tpi	824	824
Hengerszám	615	615
Adatfej	4	6
Lemez	2	3

9. táblázat

Adatok	ST225	ST238R
Kapacitás, formálatlan		
M	25,6	38,4
M felületenként	41,664	62,496
sáv	10416	15624
Kapacitás, tipikus formált (512 bájt-szektor)		
M	21,4	32,7
K felületenként	2460	2460
sáv	8704	13312
szektor	68	104
szektor/sáv	17	26
Átviteli seb. Mbit/s		
Átlagos	5	7,5
Keresési és léptetési idő ms sávtól sávig	20	20
Maximum	150	150
Fordulatszám	60	60
Hozzáférés ms	8,3	8,3
Feliratozási sűrűség bpi	9827	14740
Sávsűrűség tpi	588	588
Hengerszám	615	615
Adatfej	4	4
Lemez	2	2

10. táblázat

11. táblázat

Adatok	ST277R	ST296N
Kapacitás, formálatlan		
M	76,9	—
M felületenként	93,744	—
sáv	15624	—
Kapacitás, tipikus formált (512 bájt-szektor)		
M	65,5	85
K felületenként	4920	4908
sáv	13312	17408
szektor	156	203
szektor/sáv	26	34
Átviteli seb. Mbit/s		
Átlagos	7,5	12
Keresési és léptetési idő ms sávtól sávig	8	8
Maximum	95	70
Fordulatszám	60	60
Hozzáférés ms	8,3	8,3
Feliratozási sűrűség bpi	14902	19869
Sávsűrűség tpi	777	777
Hengerszám	820	818
Adatfej	6	6
Lemez	3	3

gyártották az elmúlt évtized slágerét, az SA400 típust. Shugart a cégtől éles viták között távozott, egy darabig szakértő volt, majd az egyik alapítója lett mostani cégének, a Seagate-nek.

Új vállalata kezdte el gyártani az első 5" méretű winchestert, az ST506-ot, amelynek csatlakozója szabvánnyá vált. Ez volt az egyik legelső olyan cég, amely Távol-Keletre, Szingapúrba helyezte át a gyártást, de a fejlesztést otthon tartotta, együttműködve a Carnegie—Mellon, a San Diego-i és a Santa Clara-i egyetemekkel.

A winchester-tárolóknál egyébként folytatták a méretcsökkenési és egyúttal kapacitásnövelési tendenciát. E tárolók mérete kezdetben 14" volt, jelenleg 3,5" méretűek, kapacitásuk pedig 5 M-ról egészen 760 M-ig nőtt.

3,5" lemezegységek

(ST125, 138, 157. *2. kép*)

Valamennyi SCSI kompatibilis, és elsősorban az IBM PS/2 sorozathoz készültek. Adataik a *8.* és *9. táblázatban* találhatók.

Adatok	ST4096	ST4192E
Kapacitás, formálatlan		
M	96	191,1
M felületenként	93,744	166,656
sáv	10416	20832
Kapacitás, tipikus formált (512 bájt-szektor)		
M	80,2	169,1
K felületenként	9216	9176
sáv	8704	18432
szektor	153	288
szektor/sáv	17	36
Átviteli seb. Mbit/s		
Átlagos	5	10
Keresési és léptetési idő ms sávtól sávig	6	5
Maximum	55	49
Fordulatszám	60	60
Hozzáférés ms	8,3	8,3
Feliratozási sűrűség bpi	9792	20078
Sávsűrűség tpi	1031	1047
Hengerszám	1024	1147
Adatfej	9	8
Lemez	5	5

12. táblázat

5" félmagas lemezegységek

(ST225, 238, 277, 296. *3. és 4. kép*)

Az első típust nem kell bemutatnunk, de adatait, ahogy a többiét is, itthon alig ismerik. Ezért a *10.* és *11. táblázatban* közöljük azokat. A választékban az ST412 és SCSI változatok is szerepelnek: az utóbbiak jele az N a típusszám után.

5" teljes magasságú lemezegységek

(ST4096, 4192.)

Az adatok a *12. táblázatban* láthatók. Az ST4096 a legelterjedtebb 80 M kapacitású ilyen egység az Egyesült Államokban. Ezek is kétféle illesztővel léteznek.

A Seagate teljes, önálló lemezrendszereket is készít. Ilyen a DISC DATA MANAGER/PLUS. Ez maximálisan tíz egységből áll, és kapacitása 640 M. Egy 80286 vagy 80386 típusú processzort tartalmaz. Elsősorban hálózati használatra ajánlják.

SIMONYI ENDRE

Spectrum-Loader

COMMODORE 64

Minden bizonnyal többeket is érdekelne a hazai két legelterjedtebb mikroszámítógép, a Sinclair Spectrum és a Commodore 64 összekapcsolása. Megfelelő interfészek és egy speciális szoftver segítségével lehetőség van a BASIC programok átadására is. Az alább ismertetett módszerrel a Spectrumhoz írt BASIC program áttölthető a C64-be, és esetleges javítások után ez a program a C64-en futtatható is.

Az elektronikus összekapcsolásra jó lehetőséget ad az RS232-es interfész, ami a Spectrumhoz kapható INTERFACE—1-ben megvan, és a C64-ben is alapjában véve megtalálható. De a C64-hez is szükséges egy kiegészítő interfész, hogy szabványos RS232-es csatlakozási felületet kapjunk. A betöltő szoftver a Spectrum által küldött BASIC listát áttranszformálja úgy, hogy a C64-en használható legyen (például a kisbetűket nagybetűkké alakítja, az utasításokat tokenizálja), és a BASIC munkaterületre tölti.

A gépi kódban írt programban *(1. lista)* az interpreter rutinjai is szerepelnek. Az egész gépi kódú rész a 49153-as ($C001) címtől kezdve a 49417-es ($C109) címig tart. A tulajdonképpeni betöltő a C0B5 és a C0ED címek között van elhelyezve, a C0FF—C106 címeken pedig az interpreter vektorának átállítása található.

A gépi kódú program betöltése és az átviteli csatorna megnyitása a *2. listán* látható két BASIC sorral végezhető el. A Spectrumból a *3. listán* látható sorral indíthatjuk a program áttöltését.

Ez a módszer jó lehetőséget ad a Spectrum programok Commodore printeren való kilistázására, ugyanis köztudottan a Spectrumhoz nagyon nehéz nyomtatót kapni.

A Spectrum-Loader program korlátai a következők. Ha az áttölteni kívánt Spectrum programban 80 karakternél hosszabb sorok vannak, azt a 80. karakternél levágja és a többit nem veszi figyelembe.

A Spectrum és a Commodore BASIC-je közti különbség miatt az áttöltés után, de a futtatás előtt egyes utasításokat és függvényeket át kell írni a C64-en (például PRINT AT, INKEY$, PAUSE stb.), különben SYNTAX ERROR hibajelzést kapunk. Ha csak a listát akarjuk nyomtatón kiíratni, az átírás szükségtelen.

PAPP LÁSZLÓ

```
C001  6C 02 03     JMP   (0302)
C004  20 B5 C0     JSR   C0B5
C007  86 7A        STX   7A
C009  84 7B        STY   7B
C00B  20 73 00     JSR   0073
C00E  AA           TAX
C00F  F0 F0        BEQ   C001
C011  A2 FF        LDX   #FF
C013  86 3A        STX   3A
C015  90 06        BCC   C01D
C017  20 79 A5     JSR   A579
C01A  4C E1 A7     JMP   A7E1
C01D  20 6B A9     JSR   A96B
C020  20 79 A5     JSR   A579
C023  84 0B        STY   0B
C025  20 13 A6     JSR   A613
C028  90 44        BCC   C06E
C02A  A0 01        LDY   #01
C02C  B1 5F        LDA   (5F),Y
C02E  85 23        STA   23
C030  A5 2D        LDA   2D
C032  85 22        STA   22
C034  A5 60        LDA   60
C036  85 25        STA   25
C038  A5 5F        LDA   5F
C03A  88           DEY
C03B  F1 5F        SBC   (5F),Y
C03D  18           CLC
C03E  65 2D        ADC   2D
C040  85 2D        STA   2D
C042  85 24        STA   24
C044  A5 2E        LDA   2E
C046  69 FF        ADC   #FF
C048  85 2E        STA   2E
C04A  E5 60        SBC   60
C04C  AA           TAX
C04D  38           SEC
C04E  A5 5F        LDA   5F
```

```
C050  E5 2D        SBC   2D
C052  A8           TAY
C053  B0 03        BCS   C058
C055  E8           INX
C056  C6 25        DEC   25
C058  18           CLC
C059  65 22        ADC   22
C05B  90 03        BCC   C060
C05D  C6 23        DEC   23
C05F  18           CLC
C060  B1 22        LDA   (22),Y
C062  91 24        STA   (24),Y
C064  C8           INY
C065  D0 F9        BNE   C060
C067  E6 23        INC   23
C069  E6 25        INC   25
C06B  CA           DEX
C06C  D0 F2        BNE   C060
C06E  EA           NOP
C06F  EA           NOP
C070  EA           NOP
C071  20 33 A5     JSR   A533
C074  AD 00 02     LDA   0200
C077  F0 88        BEQ   C001
C079  18           CLC
C07A  A5 2D        LDA   2D
C07C  85 5A        STA   5A
C07E  65 0B        ADC   0B
C080  85 58        STA   58
C082  A4 2E        LDY   2E
C084  84 5D        STY   5D
C086  90 01        BCC   C089
C088  C8           INY
C089  84 59        STY   59
C08B  20 B8 A3     JSR   A3B8
C08E  A5 14        LDA   14
C090  A4 15        LDY   15
C092  8D FE 01     STA   01FE
C095  8C FF 01     STY   01FF
```

```
C098  A5 31        LDA   31
C09A  A4 32        LDY   32
C09C  85 2D        STA   2D
C09E  84 2E        STY   2E
C0A0  A4 0B        LDY   0B
C0A2  88           DEY
C0A3  B9 FC 01     LDA   01FC,Y
C0A6  91 5F        STA   (5F),Y
C0A8  88           DEY
C0A9  10 F8        BPL   C0A3
C0AB  EA           NOP
C0AC  EA           NOP
C0AD  EA           NOP
C0AE  20 33 A5     JSR   A533
C0B1  4C 80 A4     JMP   A480
C0B4  FA           ***
C0B5  A0 00        LDY   #00
C0B7  A5 91        LDA   91
C0B9  C9 7F        CMP   #7F
C0BB  F0 33        BEQ   C0F0
C0BD  A2 02        LDX   #02
C0BF  20 C6 FF     JSR   FFC6
C0C2  20 E4 FF     JSR   FFE4
C0C5  C9 00        CMP   #00
C0C7  F0 EE        BEQ   C0B7
C0C9  C9 0A        CMP   #0A
C0CB  F0 EA        BEQ   C0B7
C0CD  C9 0D        CMP   #0D
C0CF  F0 16        BEQ   C0E7
C0D1  C9 60        CMP   #60
C0D3  30 02        BMI   C0D7
C0D5  49 20        EOR   #20
C0D7  99 00 02     STA   0200,Y
C0DA  20 D2 FF     JSR   FFD2
C0DD  C8           INY
C0DE  C0 50        CPY   #50
C0E0  90 D5        BCC   C0B7
C0E2  4C EB C0     JMP   C0EB
C0E5  EA           NOP
C0E6  EA           NOP
C0E7  C0 00        CPY   #00
C0E9  F0 05        BEQ   C0F0
C0EB  98           TYA
C0EC  AA           TAX
C0ED  4C CA AA     JMP   AACA
C0F0  A9 83        LDA   #83
C0F2  8D 02 03     STA   0302
C0F5  A9 A4        LDA   #A4
C0F7  8D 03 03     STA   0303
C0FA  20 7B E3     JSR   E37B
C0FD  60           RTS
C0FE  03           ***
C0FF  A9 04        LDA   #04
C101  8D 02 03     STA   0302
C104  A9 C0        LDA   #C0
C106  8D 03 03     STA   0303
C109  60           RTS
```

1. lista

```
1 IF A=0 THEN A=1:LOAD"SPE0",8,1
2 OPEN2,2,0,CHR$(8)+CHR$(0):SYS 49407
```

2. lista

3. lista

```
9990 FORMAT "T",1200: OPEN #4,"T"
9991 LIST #4: CLOSE #4
```

Ki ad magyarázatot?

Olvasónk, Varga Gábor küldte be az alábbi cikket:

„Határozott két- és háromismeretlenes lineáris egyenletrendszer megoldása C64-en

A program segítségével az

$a(1)\times+a(2)y=b(1)$

$a(3)\times+a(4)y=b(2)$

és az

$a(1)\times+a(2)y+a(3)z=b(1)$

$a(4)\times+a(5)y+a(6)z=b(2)$

$a(7)\times+a(8)y+a(9)z=b(3)$

alakú egyenleteket oldhatjuk meg. A gyökök értékét a Cramer-szabály adja:

$$x=\frac{DX}{D}\quad y=\frac{DY}{D}\quad z=\frac{DZ}{D}$$

ahol D a rendszer determinánsát jelenti:

$$D=\begin{vmatrix} a(1) & a(2) \\ a(3) & a(4) \end{vmatrix} \quad \text{ill.}\ D=\begin{vmatrix} a(1) & a(2) & a(3) \\ a(4) & a(5) & a(6) \\ a(7) & a(8) & a(9) \end{vmatrix}$$

A DX, DY és DZ az a determináns, amelyet úgy kapunk, hogy D-ben rendre DX-hez az első, DY-hoz a második, DZ-hez a harmadik oszlop helyére beírjuk a b(I) állandó tagokat.

Például:

$$DY=\begin{vmatrix} a(1) & b(1) \\ a(3) & b(2) \end{vmatrix}\ \text{stb.}$$

A másodrendű determinánst a

$$D=\begin{vmatrix} a(1) & a(2) \\ a(3) & a(4) \end{vmatrix}=a(1)a(4)-a(2)a(3)$$

képlet szerint, a harmadrendű determinánst pedig a Sarrus-szabállyal számítjuk ki:

$$D=\begin{vmatrix} a(1) & a(2) & a(3) \\ a(4) & a(5) & a(6) \\ a(7) & a(8) & a(9) \end{vmatrix}=a(1)a(5)a(9)+a(2)a(6)a(7)+a(3)a(4)a(8)-a(3)a(5)a(7)-a(1)a(6)a(8)-a(2)a(4)a(9)$$

A program a felhasználótól először megkérdezi, hogy két- vagy háromismeretlenes egyenletet akar-e megoldani. A választól függően bekéri a 4, illetve a 9 a(I) tagot, majd a 2, illetve 4 b(I) tagot. Ha a rendszer determinánsa nem zérus, $D\neq 0$, akkor az egyenletrendszer határozott és egy megoldása van. A program kiírja a számolt gyököket, majd felkínálja az újabb számolás lehetőségét, amit igen/nem válasszal dönthetünk el.

Ha a rendszer determinánsa zérus, $D=0$, akkor hibajelzést kapunk („a rendszer nem határozott"). A program több magyarázatot nem igényel, a részletek a listázásból kiderülnek."

```
100 REM ************************************
110 REM *                                  *
120 REM *  HATAROZOTT 2 ES 3 ISMERETLENES  *
130 REM *LINEARIS EGYENLETREDSZER MEGOLDASA*
140 REM *                                  *
150 REM ************************************
160 PRINT"■"
170 PRINT"HANY ISMERETLENES AZ EGYENLETRENDSZ.:";
180 GET TA$:IF TA$="" THEN 180
190 IF TA$=CHR$(50) THEN 220
200 IF TA$=CHR$(51) THEN 380
210 PRINT"■":GOTO 170
220 REM ***2 ISMERETLEN***
230 PRINT "■"
240 FOR I=1 TO 4
250 PRINT "A("+RIGHT$(STR$(I),1)+")=";:INPUT A(I)
260 NEXT I
270 PRINT "■"
280 FOR I=1 TO 2
290 PRINT "B("+RIGHT$(STR$(I),1)+")=";:INPUT B(I)
300 NEXT I
310 D=A(1)*A(4)-A(2)*A(3)
320 IF D=0 THEN PRINT"■A RENDSZER NEM HATAROZOTT":GOTO 580
330 DX=B(1)*A(4)-A(2)*B(2)
340 DY=A(1)*B(2)-B(1)*A(3)
350 X=DX/D:Y=DY/D
360 PRINT"■X=";X,"Y=";Y
370 GOTO 580
380 REM ***3 ISMERETLEN***
390 PRINT "■"
400 FOR I=1 TO 9
410 PRINT "A("+RIGHT$(STR$(I),1)+")=";:INPUT A(I)
420 NEXT I
430 PRINT "■"
440 FOR I=1 TO 3
450 PRINT "B("+RIGHT$(STR$(I),1)+")=";:INPUT B(I)
460 NEXT I
470 D0=A(1)*A(5)*A(9)+A(2)*A(6)*A(7)+A(3)*A(4)*A(8)
480 D=D0-A(3)*A(5)*A(7)-A(1)*A(6)*A(8)-A(2)*A(4)*A(9)
490 IF D=0 THEN PRINT"■A RENDSZER NEM HATAROZOTT":GOTO 580
500 D0=B(1)*A(5)*A(9)+A(2)*A(6)*B(3)+A(3)*B(2)*A(8)
510 DX=D0-A(3)*A(5)*B(3)-B(1)*A(6)*A(8)-A(2)*B(2)*A(9)
520 D0=A(1)*B(2)*A(9)+B(1)*A(6)*A(7)+A(3)*A(4)*B(3)
530 DY=D0-A(3)*B(2)*A(7)-A(1)*A(6)*B(3)-B(1)*A(4)*A(9)
540 D0=A(1)*A(5)*B(3)+A(2)*B(2)*A(7)+B(1)*A(4)*A(8)
550 DZ=D0-B(1)*A(5)*A(7)-A(1)*B(2)*A(8)-A(2)*A(4)*B(3)
560 X=DX/D:Y=DY/D:Z=DZ/D
570 PRINT"■X=";X,"Y=";Y,"Z=";Z
580 PRINT"■■UJABB SZAMITAS ■I/N■?"
590 GET TA$:IF TA$="" THEN 590
600 IF TA$="I" THEN 160
610 PRINT"■"
620 END
```

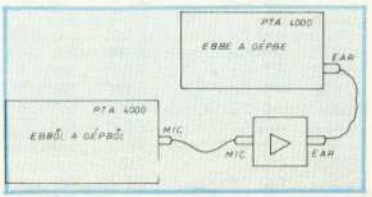
Átvizsgálva a cikkhez tartozó programot, az bizonyos kiindulási adatoknál hibás eredményt ad. Nem ad megoldást akkor sem, ha a rendszer determinánsa nem zérus.

Mi lehet ennek az oka? Milyen esetekben kapunk hibás eredményt? Hogyan tehetjük használhatóvá a programot?

S. E.

Jelátvitel két PTA 4000 között

Alkalomadtán szükségünk lehet arra, hogy egy PTA 4000 számítógépben lévő adatokat vagy programot egy másik PTA 4000-be vigyük át. Mit tehetünk ilyenkor? Az adatátvitel megoldható úgy is, hogy magnóra vesszük az adatokat és azokat innen beolvassuk a másik gépbe. A magnó kihagyásával közvetlenül összekapcsolt gépek között azonban az átvitel nem jön létre, mert a MIC jelű kapcsokon kijövő jel nem elég nagy ahhoz, hogy azt bemenő jelként a másik gép az EAR kapcson fogadhassa. Ezért egy 741-es IC-vel erősítőt építettünk, és ezen keresztül — felhasználva a gépek magnócsatlakozó kábeleit — valósítottuk meg az adatátvitelt (lásd az *ábrát*).



Az erősítőt 9 V-os telep táplálja, amelyről mintegy 2 mA-t fogyaszt. Az erősítőt, a telepet és a két csatlakozó aljzatot egy műanyag dobozban helyeztük el.

Az adatátvitel a következőképpen lehetséges. A piros dugót az erősítő MIC bemenetéhez és annak a gépnek a MIC kapcsaihoz csatlakoztatjuk, amelyikből az adatokat továbbítani akarjuk. E gépbe olyan utasítást írunk be, mintha magnóra akarnánk írni. A többi dugót nem használjuk. Egy másik kábellel összekötjük az erősítő és az adat fogadására szolgáló PTA 4000 EAR kapcsait. Ebbe a számítógépbe a magnóról való beolvasás utasítását írjuk. Először ezen a gépen nyomjuk meg az ENTER gombot, majd közvetlenül utána a másikon. Ha mindent jól kapcsoltunk össze, az adatátvitel létrejön. Az adatok természetesen az eredeti gépben is megmaradnak.

FUTÓ LÁSZLÓ

µINFORM

Rovatunkban az Apple, Atari, Commodore és Sinclair mikrók tulajdonosait feltehetően érdeklő, angol és német nyelvű cikkekről információkat adunk. A tartalomleíró szólánc segítségével.

A forráshely karaktersorozatát nyíl vezeti be, ezt a / jelig a folyóirat kódja követi (lásd táblázat). A két / jel között a megjelenési adatokat (év, hó), illetve a cikk kezdő oldalszámát szerepeltetjük. A második / jel után pedig - az esetleges másolatkérést megkönnyítendő - a cikk teljes oldalterjedelmét közöljük.

A tartalomleíró szólláncok permutálásával és alfabetikus rendezésével szerkesztett teljes anyagot az ÖMIKK "APACS" c. kiadványsorozata tartalmazza (vevőszolgálati telefon: 341-765). Lapunk ebből csak a "programlista" címszóval kezdődő részletet teszi közzé.

A folyóiratok megtekinthetők a SZÁMALK (Bp. XI. Szakasits Á. u. 68.), illetve - a x-gal jelzettek az ÖMIKK (Bp. VIII. Múzeum u. 17.) szakkönyvtárában. Másolatok a SZÁMALK-tól csekkszelvény, az ÖMIKK-tól megrendelőlevél beküldésével, vagy személyesen rendelhetők. (A másolás díja oldalanként 8 Ft.)

	A folyóirat neve	Kódja
x	64'er Magazin	64er
	Antics	anti
x	Chip Magazin	chip
x	Compute!	cute
x	Dr. Dobb's Journal	dobb
	Elektor Electronics	etor
	Happy Computer	happ
x	mc - Zeitschrift	mc
	Run (USA)	run
	Run (NSZK)	run2
x	Your Computer	your
x	ZX Computing Monthly	ZXCM

PROGRAMLISTA
ablakozas c64 (w-basic) maszk-keszites 16 uj utasitas
->run2/87.05-104/4

PROGRAMLISTA
adatbazis kezeles c64 szovegkereses (master-index) ->64er/87.05-56/5

PROGRAMLISTA
adatrendezes c64 'garbage collection' igeny csokkentese indexelt 'bubble sort' ->cute/87.05-109/2

PROGRAMLISTA
adatrendezes c64 (omega sort) (mlx-125/4) ->cute/87.05-94/2

PROGRAMLISTA
amiga jatek (rememory) emlekezet-fejleszto matrixok ->cute/87.05-40/2

PROGRAMLISTA
amiga jatek (superhirn)
->happ/87.05-54/3

PROGRAMLISTA
apple ii jatek (rememory) emlekezet-fejleszto matrixok
->cute/87.05-42/3

PROGRAMLISTA
apple ii monitor-disassembler kezeles ->mc/87.04-99/1

PROGRAMLISTA
assembler c64 masfel Kb kod/sec sebesseg tomoritett tarolas (xr1)
->run2/87.05-128/12

PROGRAMLISTA
atari st grafika demo
->mc/87.05-88/2

PROGRAMLISTA
atari st jatek (antic towers)
->anti/87.05-57/8

PROGRAMLISTA
atari st jatek (wabaduu)
->happ/87.05-117/2

PROGRAMLISTA
atari st jatek futas lassitas-lealli tas ->happ/87.05-120/1

PROGRAMLISTA
atari xl/xe filemasolo
->happ/87.05-115/2

PROGRAMLISTA
atari xl/xe grafika mat.fuggvenyek felhasznalasa (dot-hopper)
->anti/87.05-43/4

PROGRAMLISTA
atari xl/xe jatek (a-rogue)
->anti/87.05-30/6

PROGRAMLISTA
atari xl/xe jatek (poker machine)
->anti/87.05-35/5

PROGRAMLISTA
atari xl/xe jatek (rememory) emlekezet-fejleszto matrixok
->cute/87.05-44/1

PROGRAMLISTA
atari xl/xe lemezkezeles szektor editor ->cute/87.05-96/4

PROGRAMLISTA
autostart c64 egyblokkos boot-program ->run2/87.05-112/2

PROGRAMLISTA
barkacsolas c64 gyorsito lemezegyseg (1541) labkiosztas modositas parhuzamos kabel keszitese
->run2/87.05-46/6

PROGRAMLISTA
barkacsolas c64 ic-keszlet nyilvantartas (ic register)
->run2/87.05-108/4

PROGRAMLISTA
c16 jatek plus/4 (black jack)
->run2/87.05-140/3

PROGRAMLISTA
c64 (basic interrupt) segedprogramok
->chip/87.05-173/1

PROGRAMLISTA
c64 (master-copy v1.7)
->64er/87.05-53/3

PROGRAMLISTA
c64 (zoom) (87.04) szambol kimaradt anyag ->chip/87.05-174/2

PROGRAMLISTA
c64 feltores basic listavedok eltavolitasa ->run2/87.05-93/3

PROGRAMLISTA
c64 jatek (future race)
->happ/87.05-57/3

PROGRAMLISTA
c64 jatek (pick 'em up)
->run2/87.05-143/6

PROGRAMLISTA
c64 jatek (playfield) kiegeszites (robos revenge)-hez (86.12)
->happ/87.05-61/2

PROGRAMLISTA
c64 jatek (rememory) emlekezet-fejleszto matrixok ->cute/87.05-39/2

PROGRAMLISTA
c64 jatek kiegeszites a (bard's tale)-hez ->run2/87.05-150/4

PROGRAMLISTA
c64 lemezkonyvtar directory-nyomtatas lemeztasak formatum
->happ/87.05-63/3

PROGRAMLISTA
c64 nyomtatas printfox printmaster printshop formatumatalakitas printfox-hoz ->64er/87.05-69/3

PROGRAMLISTA
c64 oktatas (biologia) sejtnovekedes szimulacioja ->64er/87.05-72/2

PROGRAMLISTA
mikroprocesszor (68000) pascal compiler mnemonikus kod-outputtal
->mc/87.05-68/4

PROGRAMLISTA
ql spectrum szakkertoi rendszer 3.resz alapismeretek peldak
->zxcm/87.04-24/6

PROGRAMLISTA
spectrum 8x8-nal kisebb szines pontmezok vezerlese pelda
->happ/87.05-121/2

PROGRAMLISTA
spectrum grafika muszaki rajz 3.resz
->zxcm/87.04-70/5

PROGRAMLISTA
spectrum ikon jatekprogram keszites kephatter szoveghez
->zxcm/87.04-32/5

PROGRAMLISTA
spectrum ikon jatekprogram keszites vizszintes kepgorgetes
->zxcm/87.04-19/3

PROGRAMLISTA
spectrum segedlet basic sor-atszamozas ->zxcm/87.04-68/2

PROGRAMLISTA
sprite c128 grafika tanacsok a 'gsshape' es az 'sshape' hasznalatahoz pszeudo-sprite ->run2/87.05-118/5

PROGRAMLISTA
statisztika apple ii oszlopdiagramok felhasznaloi karakterek
->cute/87.05-70/9

PROGRAMLISTA
szovegkezeles apple ii grafika szovegiras hires ernyore
->cute/87.05-103/3

PROGRAMLISTA
vizaurite c64 helyesiras-ellenorzes egyeni szotar alapjan
->run2/87.05-114/5

PROGRAMLISTA
zene atari xl/xe (sound menu) programozasi tippek ->anti/87.05-37/4

PROGRAMLISTA
zene atari xl/xe hangkimenet-vezerles botkormannyal (sound stick)
->anti/87.05-41/2

PROGRAMLISTA
zene c64 szintetizator hangeditor 36 hangtipus-adatfile
->cute/87.05-62/7

Minsky neve jól ismert a *mesterséges intelligenciát* (angolul: Artificial Intelligence, röviden: **AI**) kutatók körében. A szerző könyve prológusában Einsteint idézi: „A dolgokat annyira kell leegyszerűsíteni, amennyire csak lehet, de nem jobban." A könyv alapkérdése az, hogy hogyan válik az önmagukban nem intelligens részek (összetevők, ágensek) együttese egyszerre csak intelligenssé. A könyv — látszólag — laza gondolatok gyűjteménye. Szinte minden egyes oldalnak címe van, a 270 gondolatoldal 30 fejezetre oszlik. E rovat gyakorlatának megfelelően most is néhány jellemző részlet felvillantásával szeretnénk felhívni olvasóink figyelmét erre az érdekes és szép könyvre.

Az egész és a részek

Mi az élet? — kérdezi Minsky. — Feldarabolod a testet, és nem találod meg a belsejében. Mi az értelem? — Feldarabolod az agyat, és ezt sem találod meg. Az élet és az értelem talán csak annyival több a részek összességénél, hogy hasztalan keressük? Minsky — a választ keresve — egy parodizált dialógusból indul ki:

Holista: Be fogom bizonyítani, hogy nincs olyan láda, melyben egy egeret fogva lehetne tartani. A láda ugyanis hat összeszegezett oldallapból áll. Az is nyilvánvaló, hogy az olyan láda, amelynek nincs valamiféle *„egérszigetelő"* tulajdonsága, az alkalmatlan egerek fogvatartására. Namármost: nyilvánvaló, hogy egyetlen oldallapnak sincs a legcsekélyebb *„egérszigetelő"* tulajdonsága sem, mivel az egér egyszerűen *lesétálhat* róluk. Ha pedig a hat lap ilyen, akkor hogyan lehetne azok együttese *„egérszigetelő"?*

Átlagpolgár: Fantasztikus! Akkor mégis mitől nem tud kiszökni az egér a ládából?

Holista: Hát nagyon egyszerűen. Bár a ládának valóban **nincs** láthatóan egyértelmű *„egérszigetelése"*, egy jó láda ezt olyan tökéletesen *„szimulálja"*, hogy ezzel becsapja az egeret és emiatt nem képes rájönni arra, hogyan kellene kiszöknie a ládából.

A paródiát félretéve, mégis, mi tartja a kis állatot a ládában? Minsky válasza: nyilvánvalóan az, hogy a láda az egér mozgását mind a hat irányban megakadályozza. Lapjait úgy állították össze, hogy a menekülés útját az **összes irányban** elzárja. Ez egyszerűen a láda titka, azaz az *„egérszigetelés"*. Ezért butaság azt feltételezni, hogy az egyes lapok *önmagukban* valamennyire is jó *„egérszigetelők"* lennének, annak ellenére, hogy ehhez a tulajdonsághoz valamennyien hozzájárultak.

Minsky szerint ugyanez vonatkozik az olyan szavakra, fogalmakra is, mint az *„élet"*, az **„értelem".** Éppen ezért ezeket nem használhatjuk az élőlények legkisebb *alkotórészeire*, mert az ilyen fogalmakat *nagyobb együtteseink* egymással és egymásra ható működésére „találtuk ki". A *„bedobozol"*-hoz hasonlóan az *„él"*, a *„gondol"* szavak olyan jelenségek leírására alkalmasak, amelyek bizonyos *kapcsolatokból, összefüggésekből* erednek. A „láda" azért nem rejtélyes senki számára sem, mert mindenki érti, hogy egy jól megcsinált láda lapjai hogyan akadályozzák meg az egér menekülését *minden* irányban. Az **„élet"** azért kezdi elveszíteni rejtélyességét — legalábbis a modern biológusok szemében —, mert már nagyon sok dolgot tudnak a sejtek vegyi anyagainak *együttműködéséről*. Az **„értelem"** még őrzi rejtélyét, mert még csak nagyon keveset tudunk arról, hogy az „értelemágensek" hogyan *működnek együtt* úgy, hogy végül is létrehozzák azt, amit létrehoznak.

Az intelligencia

Minsky könyve hetedik fejezetének elején egy külön oldalt szentel ennek a fogalomnak. Azt írja: sokan szeretnék, ha *definiálva* lenne, mi is az „intelligencia". (Így lehetne például eldönteni, hogy a növények, a kövek, a viharok, a folyók nem intelligensek-e abban az értelemben, ahogy eddig még nem is gondoltunk rá.)

Nem látszik nagyon jó ötletnek ugyanazt a szót *nagyon különböző dolgokra* használni, hacsak azok valamilyen fontos vonatkozásban nem egyeznek meg. A növények, a folyók nem oldanak meg olyan feladatokat, amelyekhez eddigi fogalmaink szerint *intelligencia* szükséges. Igen — vethetnénk közbe —, de mi az olyan különleges a *„feladatok megoldásában"?* Miért nem lehet pontosabban megmondani, miről is beszélünk?

Ez sem jó ötlet. A szerzőnek — Minsky szerint — az a feladata, hogy úgy használja a szavakat, ahogyan a *többi ember* is, nem pedig az, hogy megmondja nekik, szerinte hogyan kellene használni azokat. Ezért Minsky kijelenti: könyvének azokon a ritka helyein, ahol az *„intelligencia"* szót használja, ott ezt úgy érti, ahogy az emberek általában érteni szokták: az *intelligencia* nem más, mint *képesség* arra, hogy *nehéz problémákat* megoldjanak. (Akkor viszont meg kellene mondani, mit értünk *„nehéz"* problémán.)

Végül is Minsky oda lyukad ki, hogy: „értelmünk olyan eljárásokat tartalmaz, melyek képessé tesznek bennünket olyan feladatok megoldására, melyeket *nehezeknek* tekintünk". Az „intelligencia" az a megnevezés, amit ezekre az általunk még nem ismert folyamatokra találtunk.

Vannak, akik nem szeretik ezt a „definíciót", mert szerintük tartalma aszerint változik, hogy mennyit értünk meg a pszichológiából. De Minsky szerint ez így van jól, mert maga az intelligencia fogalma szerinte a bűvészek trükkjeihez hasonlatos. Olyan mint „a fehér foltok Afrika térképén": eltűnnek, mihelyst felfedezzük őket.

Metaforák

Csak figyelj oda jól, más mit mond neked, és mindjárt analógiákat hallasz. Az *időt* a *tér* fogalmaival írják le, mintha valami folyadék lenne, amely *elfolyik*. A jó barátokat *„közelieknek"* mondjuk. Mindennapi beszédünk tele van olyan fordulatokkal, melyeket a tárgyuktól *távoli* nyelvi jelölésekből kölcsönöztünk. Ilyenkor ezt mondják: **„metaforákat"** használunk. A metaforák az értelem különböző területei között hozzák-viszik a gondolatokat. Vannak egészen földhözragadt metaforáink: például „lépéseket teszünk" valaminek a megakadályozására. Vannak aztán meglepő, a dolgok mélyebb összefüggéseit megvilágító metaforák, melyek akár tudományos felfedezések forrásai is lehetnek. Ampère és Volta például felfedezte, hogyan lehet az elektromosságot a folyadék fogalmaival leírni (feszültséggel, áramlással). A metaforák segítségével egész új problémacsaládokat vihetünk át gondolkodásunk már ismert birodalmaiba, és ott alkalmazhatjuk rájuk a már jól kidolgozott és begyakorolt technikáinkat.

Körbe-körbe

Hacsak módunkban áll, szeretjük a körülöttünk lévő dolgokat **ok-okozat** kapcsolatokként leírni. De nem mindig tudjuk az okot és az okozatot szétválasztani. „János haza akart menni, mert fáradt volt." „János úgy érezte, hogy fáradt, mert haza akart menni." Nem lehet mindig valamilyen *„első okot"* kiválasztani. János elindulhatott, mert *egyidejűleg* fáradt volt, és úgy is érezhette, hogy már haza kell mennie. A dolgok összefüggése gyakran nem *lánc-*, hanem *hálószerű*, amelyben *hurkok* vannak. Ha — az érvelés egyszerűsítésére — a hálóinkat *láncokká* „egyesítjük", gyakran nagyon lényeges összefüggésekről, ezek felismeréséről mondunk le.

megint egy érdekes könyvet (M. Minsky: The Society of Mind. Simon and Schuster, C. 1985., 1986. 339 oldal) és ezenkívül lefordítottunk egy „szakmaközeli" Updike-írást, melyet teljes egészében közreadunk. Reméljük, sok olvasónknak megnyeri a tetszését.

Megválaszolhatatlan kérdések

— Hogyan keletkezett a világegyetem és miért?
— Mi az élet célja?
— Hogyan lehet eldönteni, melyik az igaz hit?
— Mi a jó és a mi a rossz?

Látszólag nagyon különböző kérdések. Közös bennük, hogy megválaszolásukhoz nem találunk **végső okot,** mert mindig tovább kérdezhetünk: „miért, miért?". Például, ha meg tudunk válaszolni arra, hogy valami jó-e, akkor arra is válaszolnunk kellene, hogy miért. (Emlékezzünk a kis gyerekek örökös *miért*jeire.) Mi gátolja meg ebben a felnőtteket? — A válasz kultúrafüggő. Például bizonyos kultúrkörben bizonyos kérdések *tabunak* nyilváníthatók. Minsky idézi W. H. Auden ironikus válaszát egy megválaszolhatatlan kérdésre: „Jó, azért vagyok a világon, hogy másokon segítsek. De fogalmam sincs, *mások* miért vannak a világon."

Legtöbb kultúrkör törvényekbe, vallásba, filozófiába és más intézményekbe ágyazta-foglalta a saját válaszait ezekre az egymásba visszakanyarodó (angolul: circular) kérdésekre, esetleg még kiépítette hozzá a maga tekintélyelvű, indoktrináló (besulykoló) rendszerét is. Panaszkodhatnánk, hogy dogmákkal helyettesítették a racionális érvelést, de ismerjük el, hogy cserébe sok felesleges, időrabló, sehová sem vezető vitától is megóvták a társadalmat.

Legyünk óvatosak. Ha gondolkodásunkban mindig visszatérünk ugyanoda, ahonnan elindultunk, az *nem mindig rossz.* Gondolatainkkal körbe-körbe járva gondolkodásunk *gazdagodhat:* egyre mélyebben értjük meg a dolgokat, és talán egyszer csak kiválthatjuk magunkat a bűvösnek vélt körből.

Értsünk egyet Minskyvel, és ne adjuk fel a **gondolkodást** a **gondolkodásról.** És most következzék a beígért Updike-mű.

Feladványok

1. Éjszaka van. **A,** habár **B**-vel hál, **C**-ről álmodik. **C** egy ívelt kocsifelhajtó legtávolabbi (függőleges képsíkra vetítve a látványt: az ív delelő) pontján áll. Az íves építmény valószínűleg nem más, mint álombeli tükröződése valaha közösen birtokolt házuk kocsifelhajtójának. Az asszony alakja a paradicsomszínű nyári ruhában — noha a perspektíva kicsinynek mutatja — szembetűnő. Feje hátravetve, keze csípőjén, lábai magabiztos terpeszben. Páváskodik, talán nevet is. Az egész kép telítve van asszonyi vitalitással. **A** csupa sóvárgás. **B** álma zavartalan, békésen pihen abban a szilárd tudatban, hogy **A** szereti. És valóban: **A** miatta hagyta el **C**-t — mintegy bizonyításképpen.

KÉRDÉS: **A** kettejük közül melyiket csalta meg inkább, **B**-t vagy **C**-t?

2. **A** hét utcasaroknyira lakik rendszerint használatos automata mosodájától és 3,8 mérföldnyire pszichiáterétől, akihez az út a sűrű délutáni forgalomban átlagosan 22 percig tart. Az átlagos kezelési idő — a szokásos, járulékos bevezető és levezető csevegést is beszámítva — 55 percig tart. A mosoda által favorizált felültöltős mosógépek átlagos ciklusideje 33 perc. A pszichiáter és a mosoda egy irányban van, a városból kivezető úton.

KÉRDÉS: Berakhatja-e **A** a szennyest a mosógépbe a pszichiáterhez menet úgy, hogy a nedves ruhákat közben el ne lopják?

SZORGALMI FELADAT: Ha **A** délután 3 órában állapodott meg a pszichiáterrel, ha egy-egy háztömb hosszát egynyolcad (1/8) mérföld hosszúságúnak vesszük, ha **A** két egymás utáni öblítésre állítja be a mosógépet, valamint az egész mosást a pszichiáter utánra ütemezi és ha egy-egy negyeddolláros (25 centes) érméért vásárolható öblítés-centrifugálás negyed óráig (15 percig) tart, továbbá ha egy-egy töltet két ilyen ciklust igényel, mert különben a ruha nedves marad és hazafelé menet ozmotikusan átnedvesíti cipellőjét, akkor mindezen feltételek mellett legkorábban mikor tölthet magának **A** egy pohár italt?

Kerekítsd a legközelebbi egész percre!
Oldd meg a feladatot két pohár italra!
Oldd meg háromra, belekalkulálva a mellény átnedvesedését is!

3. **A**-nak négy gyermeke van. Kettő egyetemre, kettő magániskolába jár. Az évi tandíj az egyetemen fejenként 6300 dollár, a magániskolában 4700. **A** évi jövedelme $\mathbf{n} \cdot \mathbf{n}$ háromheted (3/7) részét elviszi a szövetségi, illetve a helyi adó. Egyharmad (1/3) **C**-hez megy, aki éppen most korszerűsíti a kocsifeljárót. A teljes oktatási költség **n** öthuszonegyed (5/21) részével egyenlő. A heti egy pszichiátriai kezelés 45 dollárba kerül, a mosoda heti 1,10-be. Az egyszerűség kedvéért tekintsük úgy, hogy ezek **A** egyedüli kiadásai.

KÉRDÉS: Meddig képes **A** ezt így folytatni?

4. A borsókő köbmétere 13 dollár. Egy teherautóra 3 1/4 köbméter fér fel. **C** kocsifelhajtója 8 láb és 6 hüvelyk széles és ellipszis alakú, melynek két fókusza — két régi krokettkaróval kitűzve — 31 yardra van egymástól. A két karót összekötő vonalat középen derékszögben metsző vonal az ellipszist ettől a középponttól éppen kilenc lépés távolságra metszi — ahogy ezt az építési vállalkozó annak idején lelépte. Ő nagydarab ember, aki 48-as cipőt visel. Azt is mondja, hogy a borsókő átlagos kívánatos mélysége egy kertvárosi kocsifelhajtónál egy-másfél hüvelyk között van. Ha több, a kerék nyomot hagy; ha kevesebb, nem adja azt az elbűvölően csikorgó hangot, amelyet akkor kapunk, ha üveggolyókat pörgetünk egy kávéskannában.

Az ellipszis alakú kocsifelhajtó mellett ott van még egy ezt a Kedves útra bekötő egyenes szakasz is. Ennek a bekötőútnak a hossza úgy aránylik az ellipszis nagy tengelyéhez, mint $\sqrt{2}$ aránylik a Π-hez.

A teherfuvar alapárához hozzá kell még adni a sofőr órabérét, 10,50-et, továbbá alkalmanként annak a nagyvonalúan felajánlott sörnek az árát is, amelynek kartonja 1,80.

KÉRDÉS: Vajon miért csinálja **C** mindezt?

5. A pszichiáter úgy gondolja, hogy amit **A** észlel, az nem más, mint a **C**-től való lelki távolodással arányos növekedés. Az ún. „Trisztán-törvény" szerint azonban a vonzódás mértéke fordítottan arányos az elérhetőség mértékével, és az elérhetőség valamilyen mértékben arányos a lelki távolsággal. Ahogy az **M** lelki tömeget a távolodó perspektíva a látszólagos mértékre csökkenti, úgy nő ezzel arányosan a gravitációs vonzereje. Létezik valahol egy görbe, amelyet átlépve ez a gravitációs vonzerő túlnő a józan észen, annak ellenére, hogy a vonzás látszólagos forrása — akár a távoli csillagok látszólagos helyzete — nem más, mint illúzió.

FELADAT: Szerkeszd meg ezt a görbét! Találd meg azt a csillagtávoli pontot, ahol **A** gondolatai kezdenek elhajolni!

SEGÍTSÉGÜL: A szövegben szereplő „valamilyen mértékben" 3/7-ként számítandó át.

„Midas Törvénye": A birtoklás megszünteti az értéket — mégpedig azonnal.

6. **B** gyönyörű. Tiszta kék szemek, kék farmer miniszoknya, édes kis kék erecskék a selymes térdhajlatban. **C** gyorsan távolodik, már csak egy paradicsompiros pont a háboritatlan kék égen. **A** mind a négy gyereke ösztöndíjat kapott. A pszichiáter átköltöztette heverőjét egy diófa burkolatú, subaszőnyeges lakásba éppen az automata mosoda fölé, mindössze 22 lépcsőfoknyira a mosodától. A borsókő ára a gazdasági válság miatt drámaian esett. Gyönyörű nap ez a mai, ragyogó kék hétfő.

KÉRDÉS: Mégis valahol valami baj van. Mi lehet az?

(Fordította: KE)

From the book PROBLEMS AND OTHER STORIES by John Updike.

Copyright © 1971, 1972, 1975, 1983 by John Updike.
„Problems" originally appeared in The New Yorker.

A KGYK típusú gyalogvégjátékok

A most ismertetett program a már elemzett algoritmust egészíti ki. Használatával könnyebben s pontosabban értékelhető a pozíció. Ezért ha ezt a szubrutint a programunkba építjük, jelentősen javíthatjuk annak játékerejét. Igaz, ennek az eljárásnak a programlistája nem teljes, mert csak akkor használható, ha világosnak gyalogja van, valamint ha ez a gyalog vezérszárnyon helyezkedik el. Vagyis az a,b,c,d vonalak valamelyikén. Ennek ellenére a program használható.

Az első feltételt egyszerűen áthidalhatjuk úgy, hogy ha sötét gyalog van a táblán, akkor a szubrutint úgy használjuk, mintha az világos lenne. Ezt azzal érhetjük el, hogy a táblát 180 fokkal megfordítjuk a szubrutinhívás előtt és után is. Másrészt pedig a királyszárnyi gyalogot a középvonalra való egyszerű tükrözéssel visszavezethetjük a vezérszárnyi gyalog esetére.

A szubrutin nem teljes értékelést végez, hanem az állásokat osztályokba sorolja. Összesen harmincnyolc osztályt különböztet meg. Az *1. táblázat* azt mutatja, hogy az egyes osztályokban az értékelőfüggvényben használt képletek közül melyiket és mennyire kell figyelembe venni. A funkció(index) címszó alatt négy oszlopot találunk. E négy oszlopban vannak azok az indexek, amelyek a *2. táblázat* matematikai képleteire utalnak (1—13-ig). Egy osztályban több indexet is találhatunk, ezek egymás után fontossági sorrendben következnek.

Osztály	Funkció (index) 1.	2.	3.	4.
1	2	3	—	—
2	—	—	—	—
3	—	—	—	—
4	1	—	—	—
5	2	3	—	—
6	—	—	—	—
7	—	—	—	—
8	—	—	—	—
9	—	—	—	—
10	1	7	11	13
11	—	—	—	—
12	8	4	—	—
13	1	7	11	—
14	7	1	11	—
15	—	—	—	—
16	1	11	12	—
17	7	—	—	—
18	—	—	—	—
19	4	8	—	—
20	—	—	—	—
21	5	—	—	—
22	1	7	—	—
23	7	—	—	—
24	—	—	—	—
25	1	7	4	—
26	4	7	—	—
27	—	—	—	—
28	—	—	—	—
29	1	4	9	13
30	1	7	—	—
31	1	—	—	—
32	—	—	—	—
33	7	1	—	—
34	1	6	—	—
35	4	10	—	—
36	—	—	—	—
37	—	—	—	—
38	2	3	7	9

1. táblázat

Index	Definició	Max. vagy min.
1	pr	max
2	dist(wkf,wkr,pf,pr)	min
3	min(abs(wkf − pf),abs(wkr − pr))	min
4	abs(wkf − pf)	min
5	abs(wkf − pf)	max
6	wkr − pr	max
7	wkr	max
8	abs(wkr − pr)	min
9	abs(wkf − bkf)	max
10	abs(wkr − bkr)	min
11	abs(wkf − bkf)	min
12	wkr	min
13	wkf	max

2. táblázat

A program jelöléseiben az angol terminológiát követi. A változónevek a következő angol kifejezések kezdőbetűiből állnak: w: white (világos), b: black (sötét), k: king (király), p: pawn (gyalog), f: file (oszlop), r: rank (sor).

Ezek alapján például a wkr jelentése: „világos király sora". A pr annak a sornak a száma, amelyikben a gyalog áll, a prr értéke pedig megegyezik ezzel, kivéve azt az esetet, ha pr értéke 2. Ekkor a kettős lépés lehetősége miatt úgy kell tekinteni, mintha a gyalog már a harmadik soron lenne. Ezért a prr értéke ekkor 3.

A szubrutinban felhasználjuk a „dist" nevű függvényt, amivel már sokszor találkoztunk. Emlékeztetőül lássuk ennek definiálását:

$dist(i_1,i_2,i_3,i_4) = maximum(abs(i_1 - i_3), abs(i_2 - i_4))$.

KOVÁCS P. ATTILA

```
INTEGER PROCEDURE findrest(wkf,wkr,bkf,bkr,pf,pr);
    VALUE wkf,wkr,bkf,bkr,pf,pr;
    INTEGER wkf,wkr,bkf,bkr,pf,pr;

BEGIN
    BOOLEAN PROCEDURE bknear(file,rank);
    VALUE file,rank;
    INTEGER file,rank;
    bknear:=IF dist(bkf,bkr,file,rank)<=1 THEN true
      ELSE false;
    BOOLEAN PROCEDURE bknearer(file,rank);
    VALUE file,rank; INTEGER file,rank;
    bknearer:=IF dist(bkf,bkr,file,rank)<
    dist(wkf,wkr,file,rank) THEN true ELSE false;
    INTEGER prr,dwkp,dwkbk,wkinl,bkinc;
    BOOLEAN rpawn,opposide,between;
    bkinc:=IF abs(bkf-pf)<bkr-pr AND bkr>pr+1 THEN 1
      ELSE 0;
    IF pr=2 THEN prr:=3 ELSE prr:=pr;
    wkinl:=wkr-pr-abs(wkf-pf);
    bkinl:=bkr-pr-abs(bkf-pf);
    rpawn:= IF pf=1 THEN true ELSE false;
    opposide:= IF sign(wkf-pf)=sign(pf-bkf) THEN true
      ELSE false;
    between:= IF sign(bkf-wkf)=sign(wkf-pf) THEN true
      ELSE false;
    dwkp:=abs(wkf-pf); dbkp:=abs(bkf-pf);
    dwkbk:=abs(wkf-bkf);

findrest:
    IF dist(bkf,bkr,pf,pr)=1 AND dist(wkf,wkr,pf,
      pr)>1 THEN 1
ELSE IF stalemate(wkf,wkr,bkf,bkr,pf,pr) THEN 2
ELSE IF pr=8 THEN 3
ELSE IF cannot(wkf,wkr,bkf,bkr,pf,pr) THEN 4
ELSE IF dist(wkf,wkr,pf,pr)>dist(bkf,bkr,pf,pr)+
      bkinc AND (pr>2 OR bknearer(pf,3))
      AND bkr>prr+1 THEN 5
ELSE IF wkf=bkf AND wkr=6 AND bkr=3 AND dwkp=2
      AND pr=6 THEN 5
ELSE IF bknearer(pf,pr+1) AND NOT (bkf=pf AND
      bkr=pr+1) THEN 5
ELSE IF bknearer(pf,pr+2) AND bkr>pr AND pr<6 THEN 5
ELSE IF opposide AND bkr>pr AND wkr>pr
      AND ((wkinl=0 AND bkinl=0 AND wkr<bkr+1)
      OR (dist(wkf,wkr,pf,pr)<dist(bkf,bkr,pf,pr)
      AND wkinl>0 AND wkinl<bkinc+1)) THEN 6
ELSE IF rpawn AND wkf=pf AND wkr>pr AND bknear(3,wkr)
      THEN 7
ELSE IF rpawn AND bknearer(2,3) THEN 7
ELSE IF rpawn AND wkf=pf AND wkr>pr+2
      AND bknear(3,wkr-1) THEN 7
ELSE IF rpawn AND pr>2 AND wkr=pr+4
      AND bknear(3,pr+2) THEN 7
ELSE IF rpawn AND wkf=pf AND wkr=pr+3
      AND bknear(3,wkr-1) THEN 7
ELSE IF rpawn AND wkf=pf AND wkr=pr+3
      AND bknear(3,wkr-2) THEN 7
ELSE IF wkf=pf AND bkf=pf AND pr=3 AND bkr=2
      AND wkr=4 THEN 8
ELSE IF dbkp=4 AND (dwkp=2 OR dwkp=3) AND opposide
      AND wkr=8 AND ((bkr>6 AND pr=3) OR (bkr=8
      AND pr=2)) THEN 9
ELSE IF pr>bkr AND wkr>=bkr AND dbkp<=3 THEN 10
ELSE IF dbkp=3 AND pr<4 AND bkr=4
      AND wkr>=bkr AND dwkp=1 AND opposide THEN 10
ELSE IF bdbp=3 AND pr=4 AND (bkr=5 OR bkr=6)
      AND dwkp=1 AND opposide AND (wkr=4 OR wkr=5)
      THEN 10
ELSE IF dbkp=3 AND pr=5 AND (bkr=5 OR bkr=6)
      AND opposide AND wkr=4 THEN 10
ELSE IF between AND dwkbk=2 AND wrr=bkr
      AND prr<wkr AND pr>=4 THEN 10
ELSE IF dbkp=3 AND wkf=pf AND wkr>pr AND bkr<5
      THEN 11
ELSE IF dwkp\=5 AND dbkp=5
      AND between AND pr<6 AND wkr>6 THEn 12
ELSE IF dbkp=3 AND dwkbk\=2 AND bkr>4
      AND (opposide OR between OR wkf=pf)
      AND ((bkr=5 AND wkr>=5) OR (bkr=6 AND
      wkr>=5)
      OR (bkr=8 AND wkr>=7) OR (bkr>5 AND wkr>=6))
      AND NOT (wkf=pf AND pr<4 AND bkr<7 AND wkr<7)
      AND NOT (dwkp=1 AND pr<4 AND wkr=5
      AND (bkr<5 OR bkr=6)) THEN 13
ELSE IF dbkp=2 AND dwkbk=2
      AND wkr>=pr AND (opposide OR between OR
      wkf=pf) AND NOT (dbkp>3 AND dwkbk=2
      AND wkr>4 THEN 14
ELSE IF bknear(wkf,wkr+2) AND wkr(pr+2
      AND NOT ((pr<6 AND wkr=5) OR (pr=5 AND
      wkr<6) OR (pr=5 AND wkr=5 AND dwkp=1)
      OR (pr=wkr+1 AND abs(wkf-pf)=1)) THEN 15
ELSE IF pf=2 AND bkf=1 AND bkr=8
      AND ((wkr=3 AND wkr<7) OR (wkf=2
      AND wkr=6)) AND pr(wkr THEN 16
ELSE IF pf=2 AND bkf=1 AND bkr=7 AND wkf=3
      AND (wkr=6 OR wkr=7) AND pr<5 THEN 16
ELSE IF pf=2 AND bkf=1 AND bkr=7 AND wkf=3
      AND (wkr=6 OR wkr=7) AND pr=4 THEN 17
ELSE IF bknear(wkf,wkr+2) AND wkf=pf AND
      wkr<pr+1 THEN 18
ELSE IF dwkp=2 AND wkr>pr AND opposide THEN 19
ELSE IF wkr=bkr AND dwkp=dbkp
      AND opposide AND bkr>prr+1 THEN 20
ELSE IF wkr=bkr AND dwkp=dbkp-1
      AND opposide AND bkr>prr+2 THEN 20
ELSE IF dbkp=2 AND dwkp<=1 AND bkr<=7 AND pr<5
      AND wkr>5 THEN 21
ELSE IF wkf=pf AND pr<5 AND dbkp=2 AND bkr>6
      AND (wkr=6 OR (wkr=7 AND bkr<8)) THEN 22
ELSE IF wkf=pf AND dbkp=2
      AND ((wkr>6 AND wkr=bkr AND pr<4)
      OR (wkr>=6 AND bkr=7 AND pr<5)) THEN 22
ELSE IF dwkbk=1 AND dbkp=2 AND between
      AND wkr=bkr+2 AND ((pr=2 AND bkr=3) OR
      (pr=2 AND bkr=2) OR (pr=3 AND bkr=3)) THEN 23
ELSE IF dbkp=2 AND dwkp=1 AND opposide AND
      bkr=pr+1 AND wkr=pr+2 AND (pr=2 OR pr=4)
      THEN 23
ELSE IF dbkp=2 AND  pr=5 AND bkr=5
      AND wkf=pf AND wkr>pr THEN 24
ELSE IF bkr>3 AND dbkp=2 AND wkf=pf
      AND wkr>4 AND pr<7 AND wkr>pr THEN 25
ELSE IF bkr>3 AND dbkp=2 AND dwkp=1 AND opposide
      AND pr<7 AND wkr>pr AND ((pr=4 AND
      (wkr>=bkr OR (WKR=5 AND bkr<6) OR (wkr=5
      AND bkr=8) OR (wkr=6 AND bkr>6)))
      OR (pr=5 AND wkr=5 AND bkr=5)
      OR (pr=5 AND wkr>5 AND bkr>4) OR
      (wkr>=5 AND pr<4)) AND NOT (wkr=5 AND bkr>6
      AND pr>3)) THEn 25
ELSE IF bkr>3 AND dbkp=2 AND dwkp=2 AND opposide
      AND pr<7 AND (pr=4 OR pr=5) AND wkr>pr
      THEEN 25
ELSE IF dwkbk=2 AND (between OR wkf=pf)
      AND wkr>pr THEN 26
ELSE IF dwkp=1 AND dbkp=2 AND between AND wkr<6
      AND bkr=8 AND pr=5 AND NOT (pf=3 AND bkf=1)
      THEN 27
ELSE IF wkf=1 AND wkf=1 AND pf=2 AND wkr=6
      AND bkr=8 AND (pr=4 OR pr=6) THEN 28
ELSE IF wkr=6 AND bkr=8 AND dwkp(=1 AND NOT
      (wkf=3 AND bkf=2 AND pf=2) AND wkr=6
      AND bkr=8 AND pr=5) THEN 29
ELSE IF wkr>bkr AND opposide AND dwkp=1
      AND wkr<prr+5 AND dbkp=1 THEN 30
ELSE IF wkr>pr AND wkr<prr+3 AND dwkp=1 AND
      dbkp=1 AND bkr=wkr+1 THEN 31
ELSE IF wkf<bkf AND bkr=wkr+2 AND wkr>pr AND
      dwkp<2 AND wkf<>pf THEN 32
ELSE IF wkf=pf AND wkr>pr THEN 33
ELSE IF dwkp<2 AND wkr>pr AND wkr<=prr+3 THEN 34
ELSE IF wkr>pr AND wkr<bkr-1 THEN 35
ELSE IF wkr=pr AND wkr<5 AND bkr=7
      AND dwkp=1 AND dbkp=3 AND between THEN 36
ELSE IF wkf<bkf AND dwkp=2 AND wkr=pr AND
      wkr<6 AND bkr<8 AND NOT (pf=3 AND bkf=1)
      THEN 37
ELSE 38;
END;
```

Lassan már ötödik éve jelenik meg a Mikroszámítógép Magazin. Ilyenkor nemcsak az jut az ember eszébe, hogy „milyen gyorsan is múlik az idő", hanem az is foglalkoztatja, hogy az olvasókat menynyire érdekli az a rovat, amit ilyen hosszú idő óta szerkeszt. Egyre határozottabban érzem, hogy mindinkább élő kapcsolat és párbeszéd alakul ki a szerkesztőség és az olvasók között. Talán az e havi válogatás is igazolja ezt az érzésemet.

Jászolos Péter, Budapest

Az Olvasó írja c. rovatban olvastam egy választ Boros Péter levelére. Ebben azt írja, nagyon érdekelné az olvasók véleménye a copyrightról, erről a valóban nem is túl egyszerű jogi problémáról. Boros Péter felajánlotta, hogy szívesen közkinccsé tenné néhány program részletes leírását, térképeket, örökélet POKE-okat. Véleményem szerint ez egyáltalán nem copyrightba ütköző cselekmény, hiszen például a Spectrum-világ című kiadvány is állandóan közöl programleírásokat, játéktérképeket, POKE-okat. Nem látom a különbséget a kettő között.

Nem lévén jogász, nehezen tudok a kérdésére határozott választ adni. Elképzeléseim vannak; azt hiszem, a leírt eset éppenhogy súrolja a copyright problémát.

Kelemen Mihály, Polgár

A TVC-hez már kapható mágneslemezes tároló. Valószínűleg egyre többen fognak ilyet vásárolni. Gyorsasága miatt megvan az igény arra, hogy a már meglévő, kazettán rögzített programokat lemezre másoljuk át. Itt jön a bökkenő. Ugyanis a lemezen a névben nem állhat ékezetes betű, szóköz. Legalábbis az első nyolc karakterben. Sok program pedig ilyen (Mr. Alex, Cyrus—II Chess stb.). Ezeket átmásolni nem lehet. Jó lenne, ha olyan COPY készülne, amely kimentés előtt névmódosítási lehetőséget is biztosítana.

Kapható az Ofotértnél olyan szemüveg, amely a szemre káros sugárzást csökkenti, speciális bevonata révén. A gond az, hogy sokan nem szemüvegesek, és kényelmetlennek érzik a viselését. Másrészt, ahol többen nézik a tévét, ott egy szemüveg csak egy embert véd. Ez a helyzet az iskolákban is. Arra az intézménynek nincs pénze, hogy minden tanulónak vegyen. Pedig ebben a korban volna igazán nagy szükség rá. A javaslat: készítsenek a tévéképernyők méreteiben a készülékre rögzíthető lapokat, például tapadókorongos vagy öntapadós formában. Ebből kevesebb is elég lenne, és több ember szemét védené egyszerre.

Ami az első javaslatát illeti, szívesen látnánk ilyen programot. A másodikra pedig már egy olvasónkkal együttesen is tettünk javaslatot, hiszen az ötlet — ha ez a védőréteg tényleg olyan jó — Kolumbusz tojása. Talán a szemüveg készítői és forgalmazói is olvassák a lapot, és az ötletet megvalósítják. Egyébként az az érzésem, hogy közös javaslatunk talán üzleti szempontból nem túlságosan szerencsés; több szemüveg valószínűleg nagyobb bevételt jelent, mint egy-egy képernyő előtti védőlap. Ki tudja?

Kárpáti Péter, Veszprém

Dr. Kisapáti Edéné, szegedi olvasójuk levelére és az ön válaszára szeretnék reagálni. Legjobb tudomásom szerint a Seltron 200 Color Computer kompatibilis a VZ 200 és a Laser Color Computer 210, ill. 310 gépekkel. Így a dr. Tokodi Jenő által írt, A LASER mikroszámítógép-család c. könyv (140 forint) a Seltron gépekről is szól. Ezért tanácsolom ennek a könyvnek a megvételét. Ezer példányban adták ki, kapható a könyvesboltokban. Bár Seltron gépet még nem állt módomban kipróbálni, azt tudom, hogy mindkét gép pontosan ugyanazokat a perifériákat használja, és a Magazin 1988/1. számában Laser—Seltron bekezdéssel közöltek egy programot, Fordítós Játék címen. Noha a program BASIC-ben íródott, meg vagyok győződve a két gép kompatibilitásáról.

Sajnos nem ismerem a Seltron 200 Color Computert, így csak feltételezem, hogy ez az információ hasznos lehet az ilyen gépeket használó olvasóinknak.

Odry Miklós, Bácsalmás

Felvetődik a kérdés, hogy mikor kompatibilis egy számítógép egy másik géppel. Hazánkban a legelterjedtebb személyi számítógépek a Commodore család tagjai: a VIC—20, C116, C16, C Plus/4, C64. Kevésbé elterjedt típus ezeknek a gépeknek a testvére, a C128. Ez látszólag sok géptípus, de csodálatos a gyártó gondoskodása a felhasználókról, mert számtalan programozói fogást átvisz egyik gépről a másikra. Ez jellemzi különösen az adatfeldolgozást, ahol semmiféle eltérés nincs e géptípusok között. Aki az adatfeldolgozással akar foglalkozni, annak bármelyik megfelel ezek közül.Mivel a legelterjedtebb adatkezelő programok a C64-re íródtak, ha BASIC-ben készültek, könnyen átírhatók másik géptípusra, sőt utána futtathatók továbbra is C64-en.

Egy korábbi számban írt téves válaszom igen nagy visszhangot keltett. Azóta nemcsak leveleket, de néhány telefont is kaptam kompatibilitás ügyben. Röviden: nekem az a véleményem, hogy két gép szoftverben akkor kompatibilis egymással, ha egy adott program változtatás nélkül tud mindkét gépen futni. Ha azt — akár könnyűszerrel is — át kell írni, akkor sokféle jelzőt alkalmazhatunk a két gépre (például hogy közel állnak egymáshoz, egy családhoz tartoznak), de azt nem, hogy kompatibilisek egymással. Kíváncsi lennék, hogy ezzel kapcsolatban egyezik-e a véleményem az olvasókéval!

Nagy Róbert, Békéscsaba

Plus/4-es számítógép-tulajdonos vagyok. Több gépi kódú könyvem van, de ezekben sajnos a gépi kódbeli rajzolásról, a botkormány használatáról, a festésről alig-alig van valami. Kérem, írja meg a gépi kódú rajzolást, festést és a botkormány használatát; valamint a Mikromagazin májusi számában olvastam: a Supertape program képes más gépek — például a C64-nek — a programjait beolvasni Plus/4-en. Ez igaz?

Ami az első kérését illeti, továbbítom olvasóinknak. Ami pedig a másodikat, Simonyi Endre válasza: A Supertape egy gépfüggetlen kazettaformátum. Amelyik géphez elkészítették az író/olvasó részét, az képes a másik gépen készült programot olvasni, illetve olyan programot tud írni, amit más gépek be tudnak tölteni. Mindkét gépre van ilyen változat.

ifj. Váczi Imre, Karcag

Köszönettel megkaptam a Mikroszámítógép Magazint. Őszintén szólva, nem gondoltam, hogy önöknél ilyen gyors az ügyintézés! Sőt! Mikor feladtam a levelet, még abban sem voltam biztos, hogy válaszra méltatják. Kellemesen csalódtam.

Örülünk, és reméljük, hogy ezután is okozunk hasonlóan kellemes csalódásokat.

Végezetül közreadjuk Jéhn János szekszárdi olvasónk jól használható ötletét.

Ha nyomtatónk festékszalagja nem sérült, feslett vagy lyukas, akkor felújíthatjuk egy vastagabb ecset és fekete bélyegzőpárna-tinta segítségével. Húzzunk ki a kazettából valamennyi szalagot. Fektessük le sima felületre, s a tintába mártott ecsetet húzzuk végig a festékszalag közepén. Szélesebb szalagoknál két csíkot is húzhatunk. Ezután húzzuk be a szalagot a kazettába. Ha végigértünk, érdemes egyszer kézzel körbetekerni a szalagot, és az esetleges tintatöbbletet papírral felitatni. KOVÁCS GYŐZŐ

A MAGISZTER AJÁNLATÁBÓL

Commodore 16-ra készült programok:

Sejtautomaták
Adathordozó: magnókazetta
Járulékos hardverigény: nincs
Szerző: Kopszer Vilmos
Készítő: OCTASOFT
Forgalmazó: NOVOTRADE
Dokumentáció: Részletes (5 oldalas) leírás a kazettaborítón, magyarázatok a programban is
Leírás: Bemutatja az egyszerűbb sejtautomaták működését. A közismert Conway-féle életjátékon kívül annak általánosított formájával, és még további két sejtautomata-tipussal ismerteti meg használóját. A program menüvezérelt, kezelése egyszerű, sok magyarázattal segíti a témával kapcsolatos fogalmak megismerését, megértését.
Ára: 356 Ft

Kombinatorika
Adathordozó: magnókazetta
Járulékos hardverigény: nincs
Szerző: Szepesházi András
Készítő: OCTASOFT
Forgalmazó: NOVOTRADE
Dokumentáció: Kezelési leírás a kazettaborítón és a programban
Leírás: "A program játékos feladatok segítségével alakítja ki a felhasználóban a permutáció, kombináció és variáció fogalmát." - írja a NOVOTRADE ajánló jegyzéke. Maximálisan négy elem (szín vagy figura) összes lehetséges kombinációját kell a játékosnak a képernyőn előállítania. A program ellenőrzi a megoldást, és hibás vagy hiányos próbálkozás esetén megmutatja a helyes megoldást. Az ESC billentyű lenyomásával információt kérhetünk az aktuális feladatról.
Ára: 356 Ft

BASelastIC
Adathordozó: magnókazetta
Járulékos hardverigény: nincs
Szerző: Hernádi György
Készítő: ?
Forgalmazó: NOVOTRADE
Dokumentáció: Részletes leírás a kazettaborítón
Leírás:"BASIC programok tömörítésére és összefűzésére alkalmas segédprogramok." A két program együtt is használható.
Ára: 367 Ft

Rabló-rulett
Adathordozó: magnókazetta

Járulékos hardverigény: nincs
Szerzők: Csibra Gergely, Nagy Zoltán
Készítő: NOVOTRADE-DELTASOFT
Forgalmazó: NOVOTRADE
Dokumentáció: Részletes leírás a kazetta dobozában levő lapon
Leírás: A játéktermekből ismert félkarú rabló szelidített, számítógépes változata.
Ára: 392 Ft

TÁNYÉRTORONY
Adathordozó: magnókazetta
Járulékos hardverigény: nincs
Szerzők: Soós Gábor, Szilassi Lajos
Forgalmazó: NOVOTRADE
Dokumentáció: Részletes leírás a kazettaborítón
Leírás: A kombinatív, logikus gondolkodást fejlesztő játék. A játékosnak egy kilenc darab különböző méretű és helyzetű tányérból felépített tornyot kell a lehető legkevesebb lépésben rendbetenni, úgy hogy a tányérok nagyság szerinti sorrendben legyenek, és mindegyiknek lefelé legyen az alja. A rendbetevéshez egyetlen művelet végezhető: kezünket bármely két tányér közé csúsztatva megfordítjuk a toronynak a kezünk fölé eső részét.
Ára: 271 Ft

C64-re készült program

HEXAGON
Adathordozó: magnókazetta
Járulékos hardverigény: 2 botkormány
Szerzők: Valovics Gyula, Valovics Iván
Készítő: ANDROMEDA SOFTWARE
Forgalmazó: NOVOTRADE
Dokumentáció: Leírás a kazettaborítón, a játékszabályok az F3 gombbal játék közben is lehívhatók
Leírás: Stratégiai-logikai játék 2...4 személy részére, mely egy hatszögekből álló táblán játszható. A gondolkodási idő, maximális sebesség, a funkcionális mezők mennyisége beállítható. A program betöltése után demo-játék mutatja be a szabályokat.
Ára: 368 Ft

TVC-re készült programok:

Boszorkánykonyha (Kalandozás a kémiában)
Adathordozó: magnókazetta
Járulékos hardverigény: nincs
Szerzők: Gerlai Tiborné, Papp György
Készítő: OCTASOFT
Forgalmazó: NOVOTRADE
Dokumentáció: Leírás a kazettaborítón, a HELP billentyű lenyomására a programból is tájékoztatás kapható
Leírás: Játékos formában gyakoroltatja az általános iskola hetedik-nyolcadik osztályában tanult kémiai ismereteket. A használónak egy kémiai folyamatra kell gondolnia, a program barkchobaszerű kérdésekkel próbálja kitalálni, melyik volt az. Téves válasz esetén figyelmeztet a hibára. A HELP billentyű lenyomásával segítséget kérhetünk a helyes válaszhoz.
Ára: 454 Ft

5 LABDAJÁTÉK
Adathordozó: magnókazetta
Járulékos hardverigény: külső botkormány (1 játékos esetén opcionális)
Szerző: Tamási László
Készítő: NOVOTRADE-OCTASOFT
Forgalmazó: NOVOTRADE
Dokumentáció: Leírás a kazettaborítón
Leírás: A hagyományos tv-játék számítógépes változata. Az öt játék (falazás, páros falazás, foci, tenisz, páros tenisz) közül menüből lehet választani. Játszhat egy játékos a gép ellen, vagy kettő egymás ellen.
Ára: 392 Ft

PACK MAN
Adathordozó: magnókazetta
Járulékos hardverigény: opcionális külső botkormány
Készítő és forgalmazó: VIDEOTON Elektronikai Vállalat
Dokumentáció: Részletes leírás a kazettaborítón
Leírás: "PACK MAN, a csupafej emberke örökké éhes. Egy labirintusban bolyong táplálék után. A gonosz szellemek is ott mászkálnak, akiknek viszont PACK MAN képezi a táplálékot. Segítsünk hát kis barátunknak, hogy elkerülje ellenfeleit és jól belakmározhasson..."
Ára: 295 Ft

A könyvesbolt címe: Budapest, V. Városház u. 1.
Levélcíme: 1364 Budapest, Pf.: 52
Nyitva: Hétfőtől-Péntekig 10-18 h-ig
Szombaton 9-13 h-ig
Telefon: 382 402, 382 440

A Műszaki és Természettudományi Egyesületek Szövetsége (MTESZ) a szövetségi munka hatékonyságának javítására

PÁLYÁZATOT HIRDET

az "Igazgatási információs rendszer" IBM PC AT/XT kompatibilis mikrogépekre orientált megvalósítására

A már meglévő gépesítettséget és működő rendszereket, valamint a MTESZ területi szétlagoltságát figyelembe véve, **a rendszerrel kapcsolatos főbb elvárások:**

- a MTESZ információs rendszerének korszerűsítési koncepciója és programjának követelményei (megtekinthető, illetve érdeklődni lehet a MTESZ Tudomány- és Művelődéspolitikai Önálló Osztályon, Tel.: 534 271, 534 459, Kovács Józsefné)
- az információs igények széles körű kielégítése a MTESZ vezetése, központi szervezetei, tagegyesületei, területi szervezetei, jogi tagvállalatai és tagsága körében,
- a hivatali szervezet gazdasági munkájának segítése,
- az iroda-ügyviteli munka támogatása.

Igény a rendszer modulszerű felépítése, új elemekkel való bővíthetősége, az adatbázis összeegyeztethetősége, egységes nyelvek és módszerek alkalmazása, melyek a szakaszos bevezetés után biztosítják a hierarchikus rendszerbe integrált, automatizált információrendszer létrehozását.

A pályázatra olyan vállalkozók ajánlatait várjuk, akik a fenti tárgyban korszerű számítástechnikai módszerekkel készített referencia szoftvereket tudnak bemutatni.

A fejlesztés közös, sikeres megvalósítása érdekében a MTESZ a feltételeknek megfelelő pályázók közül előnyben részesíti a mozgókony, a körülményekhez rugalmasan alkalmazkodó szervezeteket.

A pályázat beküldésének határideje: 1988. szeptember 1.

LOGI Klub Szolnokon

A Megyei Művelődési és Ifjúsági Központban minden páros hét vasárnapján 9-13 óráig LOGI COMMODORE, a páratlan hétvégeken LOGI SINCLAIR foglalkozásokat tartunk.

A TUDOMÁNYSZERVEZÉSI ÉS INFORMATIKAI INTÉZET

előzetes megbeszélés szerint díjmentes programbemutatót tart (vidéken is) az általa forgalmazott oktatóprogramokból.

Horváth Zsuzsa 665 011/2663 mellék
vagy 813 197

Budapest, Pf.: 454, 1372

Évezredek, kontra évtizedek

„Historiam discimus. Non scolae, sed vitae discimus." – régi latin bölcsesség. Sajnos a latin nem beszélt, élő nyelv, jórészt már azokból is „kiholt", akiknek még volt szerencséjük tanulni az iskolában. Persze a fenti két mondat megfejtése nekik még nem okozna gondot, de gondolnunk kell a latint nem ismerő többségre is. Íme tehát a fordítás: „Történelmet tanulunk. Nem az iskolának, hanem az életnek tanulunk". Bizony, a történelem tanulásának vajmi kevés a haszna, ha csupán megismerjük a múlt összes nagy hadvezérét, császárát, királyát, vagy ha bebiflázzuk a hadvezérek fényesebbnél fényesebb győzelmeit. Pedig a győztesek mellett mindig voltak vesztesek is.

Ám mintha a beszédes példák mit sem érnének. Ha évezredekre visszatekintünk, a fejlődés ijesztő módon szinte csak annyi, hogy ma nem törrel és karddal, hanem szavakkal vívjuk a csatát. Vajon az évezredek csak ennyire tanítottak meg bennünket?

Nos, ha korábban nem is, mostanság igazán meg kellene tanulnunk, hogy a történelem nem puszta csatákból áll. Közben dolgozni is kellett, különben nem lett volna miből kiharcolni a nagyobb jusst (!). A nagy vezérek bölcsessége pedig sokkal inkább a korlátok, mint a lehetőségek helyes felismeréséből fakadt.

Mai ismereteinkkel érdekes volna elemezni a nagy birodalmak bukásának okai között az információátvitel korlátaiból eredő hibákat. Ma már jól tudjuk, hogy az információ milyen nagy hatalom. Hiányos piaci információk, rossz prognózisok miatt egy egész ország gazdasága tönkremehet, illetve tönkretehető.

Fontos tehát, hogy az emberiség történelmének évezredeiből mit tanulunk meg, de ma már az sem mindegy, *hogyan*. Az informatika nem évezredek, hanem évtizedek alatt futott be olyan utat, amelyről bizton mondhatjuk: egész jövőnket meghatározza. Néhány évtized alatt a számítástechnika korai dinoszauruszai „egerekké" (mouse!) szelidültek. Hatékonyságukra jellemző, hogy amíg hazánkban a közoktatásban a videoprogramot egy évtizeddel korábbi kezdés ellenére sem sikerült teljesen sikerre vinni, addig a számítógépeknek az iskolai léte ma már egyszerű történelmi tény.

E kicsit hosszúra nyúlt bevezető után sejthető, hogy e havi programelemzésünk tárgya csakis valami történelemoktató programcsomag lehet. Igen. A NOVOTRADE DELTAsoft „Évezredek" című, az általános iskola 5–8. osztályának történelmi tananyagát oktató termékét vizsgáltuk meg, illetve annak 30 moduljából egyet, mivel a modulok azonos alapelveken készültek (lásd a *táblázatot*).

Kezdjük a dokumentációval. Lényeges, hogy az ehhez a tárgykörhöz tartozó összes taneszköz – a történelemkönyvek, a munkafüzetek, a történelmi térkép, az ajánlott és kiegészítő irodalom – mind-mind megtalálható a programcsomag dokumentációjának utalásaiban. Ez óriási ismerethalmaz. Magának a programcsomagnak a használatát csak egy rövidke útmutató igyekszik segíteni, amely a bevezetőjében – helyesen – azzal kezdi, hogy a humán tárgyat tanító pedagógusokat próbálja „rávenni" a számítógép alkalmazására. Ha a kazetták gond nélkül betöltődnek, akkor a rábeszélés valószínűleg sikeres is lesz. A kazettás egység beállítását azonban sokszor a nem humán érdeklődésűek is hamar feladják. Különösen, ha a beállítás a drága tanórát rövidíti. Tanulság: órára nem szabad a gépet bevinni, ha a beolvashatóságról nem tudunk óra előtt meggyőződni, mert az később rendszerint csak bosszúságot okoz.

ÖSSZEFOGLALÓ ADATOK

Forgalmazó:	NOVOTRADE
Terméknév:	Évezredek
Szerzők:	szerzői kollektíva a NOVOTRADE DELTAsoft koordinálásával
Géptípus:	C16 (60 kf) és Plus/4
Hordozó:	kazetta vagy lemez
Dokumentáció:	rövid, alapos ismertető plusz a történelem-taneszközök
Ár:	modulonként 240,- Ft

A bevezetés a programcsomag céljaiként az alábbiakat sorolja fel: segíteni a történelmi tények, események rögzítését, az okokozati összefüggések meglátását; fejleszteni a tanulók történelmi gondolkodását, az időbeli és térbeli tájékozódási képességét; önellenőrzési lehetőséget nyújtani a tanulóknak; számszerűen értékelni az elért eredményt (százalék); játékosan, az újszerű eszköz varázsával megszerettetni a történelmet.

A programcsomag egyes moduljaihoz két-két program tartozik. Mindkettő még további két-két szintre tagolja a tárgyalt témát. Szintenként 20–30, modulonként pedig 80–100 feladattal kell a tanulónak megbirkóznia. A teljes tananyag tehát majdnem háromezer feladatot tartalmaz.

Egy-egy modulon belül a programok készítői a tananyagrészből négy szintet hoztak létre. 1. Tantárgyi minimum: a közepesnél gyengébbeknek, korrepetálásra. 2. Minimum-optimum: közepes képességűeknek, gyakoroltatásra. 3. Tantárgyi optimum: jó képességűeknek. 4. Tantárgyi maximum: témazárás gyanánt, tehetséggondozásra.

Egy-egy program anyagát a tanuló tetszőleges ideig ismételheti, gyakorolhatja, mert egy szint átfutása után ismételni lehet az adott szintet, a program végéről pedig visszakerül a vezérlés a program elejére. A programok a lehető legjobban igyekeznek segíteni az ismeretek elsajátítását. A tantárgyi maximumoknál például mindig szerepelnek a pontos források, nem kell a tanulónak nyomoznia az adatok után.

Az alkalmazott feladattípusok lényegében egyszerűek, de igen változatosak: igen-nem választások; egyszerű szelekciók (a, b, vagy c stb.); egyszerű kiegészítés (évszám, név, fogalom stb.); hibaelemzés (egy válasz hibás); relációanalízis (állításvariációk közül kell választani, ami izgalmas, de olykor elég nehéz feladat); kizárásos asszociáció (egy állításnak nincs párja); helyes időrendbe rakás; párosítás, vaktérképen tájékozódás, képfelismerés, szétválogatás, és még sorolni is sok.

Nos, az általunk megvizsgált modul gyötrése után az volt a véleményünk, hogy könnyű tárgy a történelem, ha jól tanítják. Mert nem volt nehéz 70, sőt a legnehezebbnek szánt szinten 90 százalék fölött teljesíteni (középiskolás tudással, persze). Lehet-e egy pedagógust hibáztatni ily módon: persze, jól adja fel a kérdést, amiben szinte benne van a válasz. De hát kinozni kell a diákot, vagy tanítani? Sikerélményt vagy kudarcélményt adjunk? Valószínűleg ebből is egy kicsit, meg abból is egy kicsit, de a sikerélményből inkább többet.

Eddig csak dicsértük a programcsomagot, de azért vannak benne apróbb hiányosságok is. Ha például három-három tételt párosítunk és az első kettőt jól eltaláltuk (amit a program hangjelzéssel azonnal kijelez!), bosszantó, hogy a biztos harmadik választ is meg kell még adni, mert a program nem veszi észre, hogy ez fölösleges. A jól megválaszolt kérdés után a tanuló rögtön megkapja a következő feladatot, pedig olykor jó volna eltűnődni még egy kicsit, hogy mit is kérdeztek tulajdonképpen? Valószínűleg nem tökéletes anyagismeretünknek tudható be, hogy csak egyetlen hibásnak vélhető válaszon sikerült rajtacsípni a programot, de szakmabeliek lehet, hogy hajba kapnának néhány válasz helyességén. A kakukktojás-elemzésekben található egy-egy erőltetettnek tűnő fordulat.

MINŐSÍTŐ ADATOK

Kezelhetőség:	kiváló
Teljesség:	kiváló
Dokumentáltság:	kiváló
Használhatóság:	kiváló
Ár/teljesítmény:	kiváló
Összbenyomás:	kiváló

Ezt a programcsomagot teljesen „kiváló"-nak tartjuk, mint az a minősítést részletező *táblázatból* is kiderül. Ez nem jelenti azt, hogy ne lenne érdemes továbbfejleszteni néhány éves tapasztalatszerzés után, de a mai körülmények között állításunkat megalapozottnak véljük.

A programcsomag rengeteg passzív szövegolvasásra kényszeríti a tanulót, ami indirekt módon támogatja az ismeretek elsajátítását. Ugyanakkor nagyon aprólékosan kidolgozott illusztrációk is segítik a mélyebb megértést. Az illusztrációk igen szemléletesek. Látszik, hogy készítőik maguk is nagyon élvezték a technika adta lehetőségeket. Számukra talán az lehetett a legnagyobb trauma, amikor egyszer csak be kellett fejezni a munkát.

ZSADÁNYI PÁL–
ifj. ZSADÁNYI PÁL

Herneczky István:
MACRO Assembler
(Budapest, 1988.
Műszaki Könyvkiadó — Novotrade,
77 oldal. Ára: 150,— Ft)

A „Lapozgató sorozat" új kötete hasznos segédeszköz az Intel 8088 processzorral, ennek utasításkészletével, az utasítások funkció szerinti csoportosításával kapcsolatos kérdésekhez.

Plum, Thomas:
Tanuljuk meg a C nyelvet
(Budapest, 1987.
Novotrade,
394 oldal. Ára: 350,— Ft)

A szerző a C nyelven keresztül vezeti be az olvasót a számítógépes programozásba. A könyv tartalmának megértéséhez előzetes programozási ismeretek nem szükségesek.
A kötet elősegíti az általános célú programozási ismeretek megszerzését, ahol is az elsődleges szempont a programok hordozhatósága és hatékonysága. A példákkal a C nyelv sajátosságait igyekszik illusztrálni, nem pedig speciális alkalmazási területeket bemutatni. Ugyanakkor nem mélyed el részletesen az utasítások szintaxisában sem, mert hiszen a szerző legfőképpen oktatni kíván, nem pedig felhasználói kézikönyvet írni.
A C nyelv a modern számítógépek programozásának gondosan megtervezett eszköze. Segítségével kisméretű, hatékony programok írhatók, amelyek nincsenek korlátozva egy adott számítógépes környezetre.
Az A melléklet a C nyelv tömör összefoglalóját, valamint a Unix rendszer C könyvtárát tartalmazza. A B melléklet részletesen tárgyalja a környezetfüggő eseteket, például a program lefordítását, a bájtok és szavak méretét.

Ráth György:
Programvédelmi rendszerek
(Budapest, 1987.
LSI ATSZ,
65 oldal. Ára: 65, -Ft)

A Magyarországon egyre inkább terjedő IBM és IBM kompatibilis gépek használatával kapcsolatban szerzői jogi és adatvédelmi problémák is felmerültek, illetve sok az olyan kérdés, amely a hozzáférésre, a használati jogra irányul.
A szerző ismerteti az IBM PC/XT felépítését, memóriatérképét, a sávok szerkezetét, a másolóprogramokat és a másolás elleni védelmet. Külön tárgyalja a siffrírozást és a jelszóhozzáférés védelmét.

Mocsáry Gábor:
Egyszerű elektronikai kapcsolások
Rajzolás és méretezés C64-gyel
(Budapest, 1988.
Műszaki Könyvkiadó,
160 oldal. Ára: 150,— Ft)

A szerző hazánkban elsőként állított össze egy gyakorlati példákkal magyarázó, eredeti programszervezési könyvet, melyet a kezdők és a haladók egyaránt haszonnal forgathatnak. A könyv segítségével megismerhető a programszerkesztés know-how-ja és a programszervezés menete gyakorlati példákon keresztül.
A harminc alapprogram a leggyakoribb — továbbvariálható — elektronikai kapcsolásokat tartalmazza. Valamennyi kapcsolás, valamennyi alkatrész a hazai szaküzletekben beszerezhető.

Koster, C. H. A.:
Programozás felülnézetben
(Budapest, 1988.
Műszaki Könyvkiadó,
267 oldal. Ára: 150,— Ft)

A kötet a módszeres programozás tankönyve. Mivel különböző előképzettségű olvasóknak készült, alig tételez fel matematikai ismereteket. A szerző példáiban az ALGOL—68 egyik utódjának tekinthető oktatási célú programozási nyelvet, az ELAN-t használja, melyet éppen a módszeres programozás megtanulására és gyakorlására terveztek. A kötet a felülről lefelé haladó programozással foglalkozik. Lépésről lépésre vezeti az olvasót az algoritmusok mibenlétének magyarázatától, jelölésükön keresztül az egész számok, a valós számok ismertetésén át az igazságértékek magyarázatáig. Ismerteti a szövegek jellemzőit, a vezérlési szerkezeteket, a különböző nyelvészeti alkalmazásokat, az adatállományokat és a velük kapcsolatos eljárásokat, a rekurzív algoritmusokat, a visszalépéses programozást. Végül a függelék tartalmazza az ELAN nyelvtanát és az ún. alappakkokat.
A fejezetek végén feladatot talál az olvasó ismereteinek elmélyítésére. A különféle ELAN-változatok egyúttal alternatívát is jelenthetnek a BASIC helyett, s bizonyos mértékig a Pascalt is fölülmúlják a módszeres programozás tanításában.

ADOK—VESZE

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közületeknek gépelt soronként (60 karakter) 100,- Ft, magánszemélyeknek az első sor 50,- Ft, minden további sor 20,-Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

ADOK

ATARI 800 LX számítógép eladó magnóval, szeptemberig garanciával. Érdeklődni: Marsi Gusztáv, Nagykőrös, Zsemberi Gy. u. 2. II/6. 2750

C16 számítógép magnóval, 64 kbájtos memóriabővítővel, botkormánnyal, dokumentációval és sok játékkal olcsón eladó. Siklósi Gábor, Biatorbágy, Kossuth Lajos u. 2. 2051

C16, Plus/4 programok eladók 10 Ft/db + utánvéttel. Tel.: 06/1/429-168

C Plus/4-re ill. kibővített **C16-ra** SUPERMON prg.! A gépi kódú programozást jelentősen megkönnyíti, az eredeti monitor parancsai mellett számos új funkciót tartalmaz. Szabad memória 1000-BFFF-ig, a magnó hallható stb. Kérjen tájékoztatót! Kopasz Krisztián, Szeged, Mátyás tér 1. 6725

Commodore 64-es programok nagy választékban kaphatók kazettán. Székely Tamás, Budapest, Rosenberg hp. u. 21. 1054

C64 programok eladók, kazettánként 40-50 program. Egy kazetta 400 Ft + utánvét. A programok külön is megvehetők, 10 Ft/db. Listát előre küldök. Válaszokat a következő címre kérem: Dukán Zoltán, Sopron, Laktanya u. 20. 9400

Commodore 64-hez GYORS-HÁTTÉRTÁR cartridge. Kapacitása 2-31 kbájt. A GYORS-HÁTTÉRTÁR-ba maximum 7 db, célszerűen gyakran használt program vihető be. A gép bekapcsolása után menüvel jelentkezik be, és gombnyomásra a kiválasztott program azonnal fut. Javasolt programcsomagok: Turbo tape, Assembler,Monitor, Supergrafik, Help plus, Turbo másoló. File másoló 1699 Ft és Turbo tape 699 Ft.
Hozott programok elhelyezése a GYORS-HÁTTÉRTÁR-ban. Trompler László, Budapest, Attila u. 22. 1201. Tel.: 287- 493 este.

C64-re színvonalas programokat (86-87) reális áron adok-veszek és cserélek. Listát kérek és küldök. Programokat lemezen és kazettán is adok, a cserélés kizárólag lemezen lehetséges. Kató János, Mezőhegyes, Cukorgyári ltp. 2. 5820

C64 + VC 1541 + datasett + 2 db joy + 80 db lemez fiatal programokkal + 7 db Data-Becker könyv + 1001/1,/2 + Spectrum játék és pr. könyvek 35 000 Ft, MPS 802 nyomtató 7 500 Ft. Zöld monitor 7 500 Ft - eladó. Pásztor György, Mád, Batthyány u. 65. 3909

Commodore 64-es számítógép floppyval, 3 cartridge-vel és 101 db lemezzel (kb. 800 program) sürgősen eladó. Az árajánlatokat levélben várom. Bielik Norbert, Budapest, Bartók Béla u. 3/C. 1225

Commodore 128 + 1541 drive eladó, külön is! Árajánlatokat a következő címre kérem: Lenhardt Tamás, ADT, Győr, Munkásőr u. 75. 9023

Commodore VC20 felhasználói és játékprogramok eladása. Kérjen tájékoztatót! Levélcím: Juhász György, Salgótarján, Pf.: 157, 3100

Commodore, Enterprise, IVC és Atari számítógépekhez HW cikkek széles választékát kínáljuk. Joystickok 450 Ft-tól kaphatók. Fényceruza 2000 Ft (C16-hoz és Plus/4-hez is). Válaszboríték ellenében elküldjük katalógusunkat. COMPUTEAM GM, Kaposvár, Berzsenyi u. 32. 7400

LACISOFT BASIC Plus/4-re! BASIC kiterjesztés, 15 új utasítás látványos képernyőkezeléshez (négykarakteres kifírás, scrollok, képcserék). Ára kazettán: 500 Ft. Tape Copy + Nova Turbo (200 Ft). "Novaload" kompatibilis másoló + Turbo (C16-on is egyszerre 12 384 bájt másolható). SZT-BT-TL SOFTWARE CO.;

Magas ár: KC 87/3

A Robotron Kombinát az idén márciusban megjelentette a KC 87/4 típusú gépét, amely jobbára csak külső megjelenésében különbözik elődjétől, a KC 87/3-tól. Az NDK-ban gyártott gépet U880 mikroprocesszorral látták el, amely funkcionálisan a Z80-asnak felel meg. A KC 87/4 teljesítménye a ZX-Spectruméhoz hasonló. A ROM-jában elhelyezett BASIC értelmező vezérlése segítségével üzemeltethető. Ára magas, a kiépítettségétől függően 3000-4000 márka, és ráadásul a gép csak ritkán kapható.

A KC 87 mikroszámítógép két perifériájával, a televízióval és a háztartási magnetofonnal

Pénzkímélő kártya

A pénzforgalom gyorsítására és a takarékszolgálat egyszerűsítésére egy teljesen újszerű szolgáltatás, az aktív memóriakártya bevezetését tervezi a Magyar Posta. A szolgáltatást alaposan előkészítik és hosszabb ideig kísérletképpen alkalmazzák, hogy a lakosság megismerhesse és kipróbálhassa. Az egésznek az alapja az aktív memóriakártya, egy olyan kis lap, amely több ezer adat — köztük a takarékbetétkönyvben lekötött összeg — tárolására képes.

Az 1989-ben kezdődő kísérlet színhelyéül egy közepes nagyságú várost, Egert választották, ahol kb. tízezer takarékbetétkönyvtulajdonos van. A város postahivatalait számítógépekkel szerelik fel, és valamennyit egy számítóközponttal kötik össze. A városban több ponton elhelyeznek majd kártyaolvasó berendezéseket, amelyek segítségével lehetővé válik, hogy a takarékkönyvből bárhol pénzt lehessen kivenni.

A számítógépes hálózatba a tervek szerint több telefonkészüléket is bekapcsolnak, és ezekkel a memóriakártya segítségével — pénz nélkül is — telefonálhatnak a betétkönyv terhére. Akinek nincs takarékkönyve, meghatározott értékű telefonkártyát vásárolhat, illetve használhat. Több kereskedelmi céget is szeretnének bevonni a kísérletbe, hogy az adott boltokban kártyával vásárolhassanak a vevők.

A kísérlet előreláthatólag egy évig tart. Ha a posta tapasztalatai kedvezően alakulnak — no meg ha lesz a kísérlethez szükséges elegendő telefonvonal —, az egész országban általánossá teszik e pénzkímélő módszert.

-CSERÉLEK

Szabolcsi Tivadar, Kaposvár, Szalma I. köz 5. 7400. Kérjen tájékoztatót!

Magyar nyelvű dokumentáció eladó az alábbi programokhoz: PRINTFOX, CHARAKTERFOX, GEOS. Cím: Honti József, Csákvár, Május 1. u. 11. 8083

Sinclair Spectrum (128 k) eladó. Ár: 19 900 Ft. Érdeklődni lehet: 06/27/41-005 ill. Róth Gábor, Budapest, Kléh István u. 9. 1126

Sinclair ZX-Spectrum (48 k) gépemet sürgősen el szeretném adni. A géphez adok még interfészt, joystickot, 500 programot és sok szakkönyvet (beépített resetgomb). Irányár: 18 000 Ft. Érdeklődni lehet: Selyem Attila, Lengyeltóti, Fonyódi u. 51. 8693

Spectrum (1987-88-as) programokat adok-veszek-cserélek. Árpási Zsolt, Mezőhegyes, Sziklai S. 1tp. 5/B. 5820

Spectrum számítógéphez saját fejlesztésű TOTÓ tippkészítő program leírással eladó. Adatbevitel a Sportfogadás táblázataiból. Ár 250 Ft + kazetta. Hári András, Nagykanizsa, Vörösmarty u. 55/a. 8800

VC20 számítógép 1530-as magnóval eladó. Árajánlatokat levélben kérek. Polgárdi Péter, Lengyeltóti, Kert u. 38. 8693

ZX-Spectrum Plus (új), ZX Interfész II-vel, 2 db Quick Gun I. joystick, 10 db szakkönyv, 650 db játék és 250 db felhasználói programmal (22 000 Ft-ért) és Primo A-64 számítógép, 3 db szakkönyvvel és 10 db játékprogrammal 4000 Ft-ért eladó. Kápolnási János, Veszprém, Felszabadulás u. 63/A. 6/38. 8200

ZX-Spectrum + 128 k sürgősen eladó 15 000 Ft-ért. Sári Gábor, Szolnok, Mátyás K. u. 10. fsz.2. 5000

VESZEK

ZX-Microdrive-ot Interfész 1-gyel összekötő szalagkábelt vennék és ZX-Spectrumra (48 k) programot cserélnék. Listát kérek! Gombos Bertalan, Bonyhád, Perczel kert 13. 7150

CSERÉLEK

C16-os (64 k) gépemet magnóval, joystickkal elcserélném Spectrumra, ill. egyéb számítógépre, vagy Commodore perifériákra. Virág Gábor, Soltszentimre, Akácfa u. 13. 6223. Tel.: 78/45-373

C16, 116 és Plus/4-es programokat cserélek kazettán. Listát kérek. Plenter Róbert, Nyíregyháza, Toldi u. 42. 4400

Márkatársak figyelem! Commodore 16, Plus/4-es programokat cserélek. Keresem a Godzilla nevű programot. Címem: Németh Tamás, Dunaújváros, Alkotás u. 10. II/1. 2400

C64-es felhasználói és játékprogramokat, dokumentációkat cserélek lemezen. (Pl.: GEOS, MAX FORTH, OXFORD PASCAL, COMAL, ACE, GAME MAKER, HI-EDDI+, PROFI-ASS, URIDIUM, COLOSSUS CHESS...) Setény János, Békéssámson, Vöröskatonák u. 97. 5946

C64-es programokat cserélek kazettán. Listát kérek! Tóth Zoltán, Tata, Akácfa u. 51. 2890

Enterprise programokat cserélek. Listát kérek! Sziderov Kraszimir, Debrecen, Batthyány u. 9. III/4. 4024

Szinvonalas Spectrum programokat cserélek. Válaszokat listával kérek. Keresem a Battle for Midway c. programot. Szirovicza Ernő, Szeged-Szőreg, Szerb u. 30. 6771

A PLATINE 64 prg. ismertetőjéért Epromégető dokumentációt adok. A 2716, 2732, 2764, 27128, 27256 típusok (NMOS/CMOS) programozásához. Illés István, Cikó, Perczel Mór u. 44. 7161

Adok: 13 cm átmérőjű, Newton rendszerű, tükrös csillagászati távcsövet állvánnyal. **Kérek:** számítógépet (ZX81 kivételével). Kalina Pál, Békéscsaba, Bezerédi u. 22. 5600

Ez a rovatunk **KODEX 2000** szövegszerkesztővel készült.

Pontvadászat

Új rejtvényt indítottunk útjára lapunkban, remélve, hogy elnyeri olvasóink tetszését. Minden számunkban két feladatot közlünk: az első logikai, matematikai tudást, a második számítástechnikai alapismereteket is igényel.

A feladatok után közöljük az elérhető maximális pontszámot. A részmegoldásokat is pontozzuk.

A pontgyűjtést, vagyis a pontvadászatot az esztendő végén zárjuk. A legjobb tíz versenyző nevét magazinunkban közzé is tesszük, ők lesznek azok, akik könyvutalványt is kapnak.

A helyes megoldások a feladatok közreadása után két lapszámmal később jelennek meg, így a pontvadászoknak jut idejük a gondolkodásra.

Beküldési határidő: 1988. szeptember 15.
Címünk: Mikroszámítógép Magazin Szerkesztősége 1371 Budapest, Pf. 433.

Az 1988/2. szám 1. feladatára hatvan, 2. feladatára ötvenkilenc megfejtés érkezett. Ezek közül az 1. feladat megfejtési százaléka (a maximális 4 pontot véve 100 százaléknak) 92,9 százalék, a 2. feladaté (a maximális 5 pontot véve 100 százaléknak) 91,9 százalék volt.

Dr. Hoffmann Tibor

1. feladat

Tekintsünk az

$ax^3-(a^2+3a-1)x^2+(b+3a^2-a)x-ab=0$

egyenletet. *Határozzuk meg* az *a* és *b* paraméter azon nem nulla értékét, melynél mindhárom gyök egymással egyenlő, és *állapítsuk meg* ezt a gyököt. (3 pont)

2. feladat

Az 1988/6. szám 2. feladatának a játékára *adjunk olyan stratégiát*, melynek segítségével megfelelő rendszer szerinti kérdezéssel ki tudjuk találni, illetve a gép ki tudja találni a tárolt számot. (8 pont)

Az 1988/6. szám 2. feladatában képzett szubrutinok felhasználásával (melyeknek a megoldását ebben a számban közöljük) *írjunk egy szubrutint* a tárolt szám megállapítására, miután megadtunk egy próbaszámot. (10 pont)

Az 1988/6. szám feladatainak megoldása

1. feladat

$x \neq 0$ és $y \neq 0$.

Vezessük be a

$z=xy$ és $v=\frac{x}{y}$

változókat. Ezekkel

$z-v=\frac{3}{4}$ és $z-\frac{1}{v}=\frac{4}{3}$.

A két egyenlet kivonásával rendezve *v*-re a következő egyenletet kapjuk:

$v^2-\frac{7}{12}v-1=0$,

melyből

$v_1=\frac{4}{3}$ és $v_2=-\frac{3}{4}$.

Ezekhez

$z_1=\frac{25}{12}$ és $z_2=0$

tartozik. Az utóbbi nem ad megoldást. A maradó megoldás tehát:

$xy=\frac{25}{12}$, $\frac{x}{y}=\frac{4}{3}$,

ahonnan

$x=\frac{5}{3}$ és $y=\frac{5}{4}$.

(4 pont)

2. feladat

Legyenek az ötelemű **IA** tömb elemei a gépben már tárolt kitalálandó szám számjegyei sorban, és a szintén ötelemű **IB** tömb legyen a játékos próbaszámának a számjegyeit tartalmazó tömb. Ekkor a következő szubrutinokat írhatjuk le például FORTRAN nyelven:

Az IB-ben elhelyezett *5* számjegyből képezze az IPSZM(1B) az *ötjegyű* számot:

```
              FUNCTION IPSZM(IB)
              DIMENSION IB(5)
C             AZ IB 5 ELEMŰ TÖMBBEL MEGSZERKESZTI
C             A SZÁMOT IPSZM-BE
              I=5
              IPSZM=Ø
              DO 1 J=Ø,4
              IPSZM=IPSZM+IB(I)*1Ø**J
1             I=I+1
              RETURN
              END
```

Az adott K *5* jegyű egész számból képezi az *5* számjegyet az IP tömbben a következő szubrutin:

```
              SUBROUTINE BONT(K,IP)
              DIMENSION IP(5)
C             A K ÖTJEGYŰ SZÁM JEGYEIT SORBAN ELHE-
              LYEZI
C             AZ IP TÖMB ELEMEIBEN
              LOGICAL HIBA
              IF(K.LT.1ØØØØ)GOTO 3
              X=K/1ØØØØ
              DO 2 J=1,5
              X=X*1Ø
              IP(J)=INT(X)
              DO 4 II=1,J-1
              IF(IP(J).EQ.IP(II))GOTO 3
4             CONTINUE
2             X=X-IP(J)
              RETURN
3             HIBA=.TRUE.
              END
```

(4 pont)

Végül a kétjegyű szám kérdező szubrutinja:

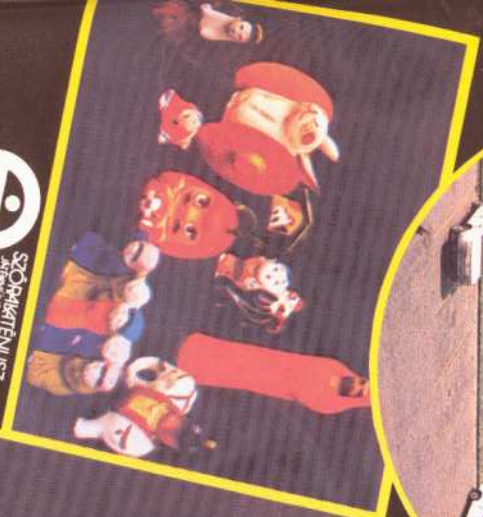
```
              FUNCTION KERD(IA,IB)
              DIMENSION IA(5),IB(5)
C             A PRÓBASZÁM ÉS A KERESETT SZÁM EGYEZŐ
C             SZÁMJEGYEINEK A SZÁMÁT TÍZZEL SZOROZ-
              VA HOZZÁADJA
C             AZ EZEK KÖZÜL MÁR A HELYÜKÖN LEVŐ SZÁ-
              MOK SZÁMÁHOZ
              K=Ø
              II=Ø
              DO 5 I=1,5
              DO 5 J=1,5
              IF (IA(I).NE.IB(J))GOTO 5
              IF (I.EQ.J)II=II+1
              K=K+1
5             CONTINUE
              KERD=1Ø*K+II
              RETURN
              END
```

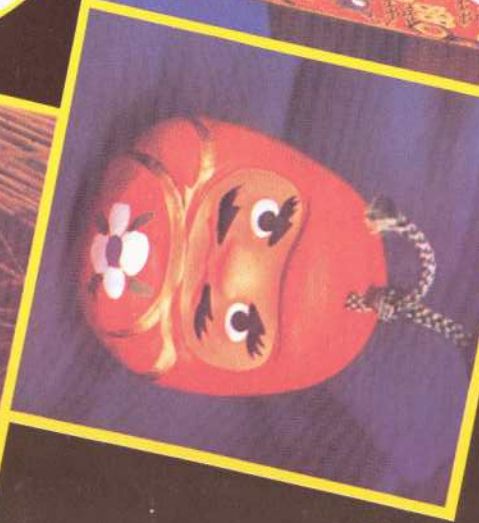
(8 pont)

A közölt megoldást és a feladat egy részét beküldte Nagy D. István (Csikszereda, Románia).

SZÓRAKATÉNUSZ

JÁTÉKMŰHELY ÉS MÚZEUM





Japán gyermekjátékok — elektronika nélkül

Gondolatok a Játékmúzeumban
(30. oldal)

1988. II. félévtől bővített szoftverrel kapható.

KODEX 2000

Az 1987. évi tavaszi BNV-n díjat kapott KODEX 2000 szövegszerkesztő berendezés kurzororientált, dinamikus képernyőformázó és menüvezérelt rendszer. Kezelése egyszerű, kényelmes és nem igényel semmiféle számítástechnikai képzettséget.

Fejlesztette:
BME Villamosmérnöki Kar, Folyamatszabályozási Tanszék,
Budapest, Műegyetem rkp. 3/9. 1111
Tel.: 453-500

Gyártja:
KONTAKTA Alkatrészgyár Ózdi Gyáregysége,
Ózd, Bolyki főút 82. 3600
Tel.: 47/12-555

Forgalmazza:
MIGÉRT Író- és Számológéposztály,
Budapest, Dimitrov tér 14. 1093
Tel.: 175-081

Szervizét ellátja:
ITV KODEX Szakszervize,
Budapest, Báthori u. 5. 1054
Tel.: 121-090